Ekmelily - Notation of Microtonal Music

Ekmelily is an extension for LilyPond that supports variable accidentals and key signatures for the notation of microtonal music in several tunings.

It currently supports the following tunings:

- Equal temperament: 12, 19, 24, 31, 36, 48, 53, 72, 96 EDO
- Just intonation: 5-limit JI

It introduces predefined and custom notation styles. Each style describes a set of accidental symbols for the alterations, up to the five-quarters-tone at most.
Furthermore, it defines own note names based on the names for semi- and quarter-tones given in LilyPond.

Ekmelily requires LilyPond version 2.24 or higher.

This documentation uses the Ekmelos font for all SMuFL glyph.

20 February 2026

Contents

Author and License

Ekmelily was written by Thomas Richter, thomas-richter@aon.at

Usage

Ekmelily on GitHub

Install from folder `ly` the file for the desired tuning as well as the main include file `ekmel-main.ily` .

Add the following lines near the top of your LilyPond input file, with `\include` for the desired tuning.
Everything else is optional and the order of the commands after `\include` is not significant.

```
ekmFont = "FONT"
\include "ekmel..."
\ekmStyle STYLE
\ekmUserStyle USERSTYLE #'((ALTERATION ELEMENT ...) ...)
\language "LANGUAGE"
```

Ekmelily + Esmuflily

See Esmuflily for the support of other SMuFL music symbols (clefs, note heads, flags, etc.) and how to combine both for a full SMuFL support.

Lookup Tables

- See the Lookup tables with all accidentals and note names supported by Ekmelily,
- or create the desired lookup tables with LilyPond using `table.ly` .

Fonts

Ekmelily requires a font for accidentals defined by code point, character literal, or string. This applies to all accidentals in the predefined notation styles. Each of them is a SMuFL recommended character or Ekmelos specific optional glyph (starting at code point U+F600).

Ekmelily uses Ekmelos by default.
Another font can be selected, either with the variable preceding the include file

```
ekmFont = "FONT"
```

or with the command line option (produces a warning "no such internal option" which can be ignored)

```
-dekmfont=FONT
```

Note that both, Ekmelily and Esmuflily select a font that way. A separate font for accidentals only can be specified with the `font-name` property.

```
\override Accidental.font-name = #"FONT"
```

The glyphs from LilyPond's Emmentaler font can be used with markup in a custom notation style. See the example `lilysingle`.

Drawing paths

Ekmelily supports drawing paths instead of font glyphs, which allows to produce stand-alone SVG output.
A trailing `#` in FONT switches to drawing paths which effects all accidentals defined by code point or character literal, in particular, all accidentals in the predefined notation styles. It includes the file `FNAME-paths.ily` if available, where FNAME is the font name in all lowercase. This file must provide the Scheme procedure `ekm-path-stencil`.
Such a file can be generated with `pathtable.py` . See `ekmelos-paths.ily` for the Ekmelos font.

Note that paths do not support:

- Spaces and other glyphs without a contour.
- Font features like stylistic alternates and ligatures.
- Side-bearing.

Tunings

Ekmelily supports different tunings available as separate include files. Each provides its own set of languages and predefined notation styles. The first language and notation style specified in the following table is the default in the respective tuning. Some languages have alias names (in parentheses).

Tuning	Include file	Languages	Notation styles
12	`ekmel-12.ily`	`nederlands english deutsch català (catalan) español (espanol) italiano français português (portugues) norsk suomi svenska svenska_ny vlaams`	`std sag msag`
19	`ekmel-19.ily`	`nederlands english deutsch català (catalan) español (espanol) italiano français português (portugues) norsk suomi svenska svenska_ny vlaams`	`std sag msag`
24	`ekmel-24.ily`	`nederlands english deutsch català (catalan) español (espanol) italiano français português (portugues) norsk suomi svenska svenska_ny vlaams`	`stc stz go stvt arrow sag msag arabic persian four haba bl`
arabic	`ekmel-arabic.ily`	`italiano arabic`	`arabic helmakam`
31	`ekmel-31.ily`	`nederlands deutsch español (espanol) français italiano português (portugues)`	`std sag msag stz sth`
36	`ekmel-36.ily`	`nederlands english deutsch norsk suomi svenska`	`go arrow sag msag wys bos haba`
48	`ekmel-48.ily`	`nederlands english deutsch`	`sag msag gostz`
53	`ekmel-53.ily`	`makam thm ktm english number`	`aeu aeuek thm sag dia`
72	`ekmel.ily`	`nederlands english deutsch norsk suomi svenska`	`arrow rhm rhmk sims hesse sag msag wys gostz gostc bos fern haba`
96	`ekmel-96.ily`	`nederlands english deutsch`	`sag msag persian om`
5ji	`ekmel-5ji.ily`	`nederlands`	`sag msag he`

Note: `ekmel-arabic.ily` is a variant of `ekmel-24.ily` for Arabic scores, like LilyPond's `arabic.ly` and `hel-arabic.ly` but with the correct accidentals U+ED30 - U+ED38. It also supports Arabic maqamat (keys).

Languages and Note names

Each tuning provides one or more languages for note names, which can be selected with the command

```
\language "LANGUAGE"
```

If LANGUAGE is not supported by the respective tuning, the default language is selected.

See the Lookup tables for all note names supported by Ekmelily, or create lookup tables with `table.ly` .
The note names are based on the names for semi- and quarter-tones given in LilyPond.

Alteration names

An alteration name is the accidental suffix of a note name in the selected language. It can be specified as a symbol in custom notation styles and in commands like `\ekmelic-char` alternative to a rational number. This is useful, in particular, to distinguish enharmonically equivalent accidentals.
Note: All languages define note names with accidental suffixes, except for `ktm` in tuning 53.

Enharmonical equivalence

Some notation styles support two distinct, enharmonically equivalent accidentals, e,g, `arrow`, `rhm`, and `sims` for the one-quarter-tone and the three-quarters-tone. Therefore, Ekmelily defines two note names each, like `cih` and `ciseh` in "nederlands", or `cqs` and `csaqf` in "english". However, LilyPond does not support different accidentals for the same alteration. As a workaround, the combined note names like `ciseh` and `csaqf` have slightly differing alterations (+1/1024) and therefore cause inaccurate MIDI output.

NoteNames context

Ekmelily supports the `NoteNames` context and its properties. Note names can be drawn in any supported language by setting the `printNotesLanguage` property. Else, the language selected for music entry is used. Accidentals are drawn with the command `\ekmelic-char-text`.

The format can be specified with the `printAccidentalNames` property. The first three values in the following table are equivalent to LilyPond's normal behaviour.

`#t`	Scale name and accidental (default)
`#f`	Scale name
`'lily`	Note name
`'all`	All alias note names stacked vertically
`'alteration`	Alteration name
`'fraction`	Scale name and fraction of alteration
`'accidental`	Accidental

Predefined Notation Styles

Each tuning provides one or more predefined notation styles, which can be selected either with the command

```
\ekmStyle STYLE
```

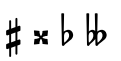
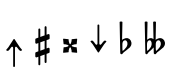
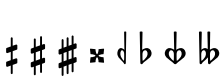
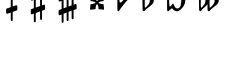
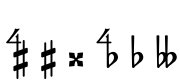
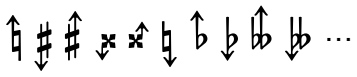
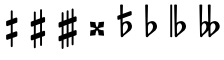
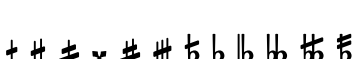
or with the command line option (produces a warning "no such internal option" which can be ignored)

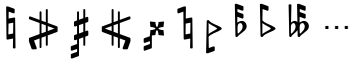
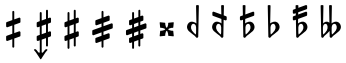
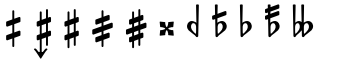
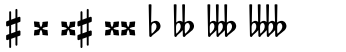
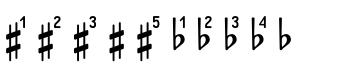
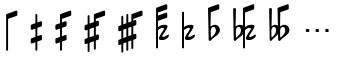
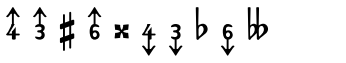
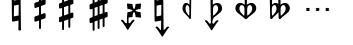
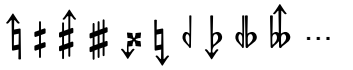
```
-dekmstyle=STYLE
```

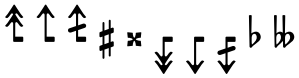
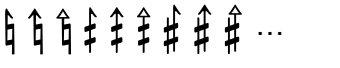
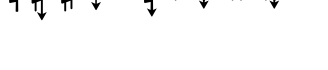
If STYLE is not supported by the respective tuning, the default notation style is selected.

All these styles require a SMuFL compliant font. All optional glyphs (starting at code point U+F600) are private supplements of Ekmelos.
Tuning N denotes the default style in this tuning.

See the Lookup tables for all accidentals in these styles, or create lookup tables with `table.ly` .

Style	Describtion	Tunings	Accidentals
`std`	Standard	12 19	
	Standard	31	
`stc`	Stein / Couper	24	
`stz`	Stein / Zimmermann	24 31	
`stvt`	Stein / Van Blankenburg / Tartini	24	
`sth`	Stein / Half flat Uses the optional glyphs U+F612 - U+F613 for the semi-flats.	31	
`bl`	Blackwood Uses the optional glyphs U+F610 - U+F611.	24	
`four`	Digit 4	24	
`go`	Gould	24 36	
`arabic`	Arabic See `ekmel-arabic.ily` for Arabic maqamat.	24 arabic	
`helmakam`	Hel Makam Uses the optional glyph U+F61B for the slashed double-flat.	arabic	

Style	Describtion	Tunings	Accidentals
`persian`	Persian	24	
	Persian Uses the optional glyphs U+F676 - U+F677 and U+F76C - U+F783.	96	
`aeu`	Arel-Ezgi-Uzdilek Uses the optional glyph U+F619 for the reversed slashed flat. See LilyPond's `turkish-makam.ly` for keys.	53	
`aeuek`	Arel-Ezgi-Uzdilek Equals `aeu` but does not use the reversed slashed flat, i.e. eksik-bakiye = koma.	53	
`dia`	Diatonic Uses the optional glyphs U+F61C - U+F61D for the quadruple sharp and flat.	53	
`thm`	Turkish folk music	53	
`arrow`	Arrow	24 36 72	
`bos`	Bosanquet commatic	36 72	
`haba`	Hába	24	
	Hába Uses the optional glyphs U+F660 - U+F670.	36 72	
`wys`	Wyschnegradsky	36 72	
`fern`	Ferneyhough	72	
`gostc`	Gould / Stein / Couper	72	
`gostz`	Gould / Stein / Zimmermann	48 72	
`hesse`	Hesse Uses the optional glyphs U+F606 - U+F60B for the degrees within the semitone.	72	

Style	Describtion	Tunings	Accidentals
`rhm`	Richter Herf / Maedel Uses the optional glyphs U+F600 - U+F605 for the degrees within the semitone.	72	
`rhmk`	Richter Herf / Maedel kink arrows Equals `rhm` but with kink arrows. Uses the optional glyphs U+F784 - U+F789 for the degrees within the semitone.	72	
`sims`	Sims	72	
`om`	OpenMusic Uses the optional glyphs U+F758 - U+F75F.	96	
`he`	Extended Helmholtz-Ellis	5JI	
`sag`	Sagittal	12 19 24 31 36 48 53 72 96 5JI	
`msag`	Mixed Sagittal Uses the large double sharp U+E47D. Hence it is different from `std` even for 12-EDO.	12 19 24 31 36 48 72 96 5JI	

Universal Notation Styles

These notation styles are available in all tunings. They do not define any accidentals, only the special symbols (`default`, `leftparen`, `rightparen`), so that the default accidental is drawn for each alteration.

Style	`default`	`leftparen`	`rightparen`
`void`	Nothing (empty markup)	`#xE26A`	`#xE26B`
`alteration`	Alteration value such as $\frac{1}{4}$	`"("`	`")"`
`alteration-slash`	Alteration value such as 1/4	`"("`	`")"`
`step`	Step of the alteration (integer)	`"("`	`")"`

The special symbols of `void` are also defined in every predefined notation style.

Alteration values are drawn with the command `\ekmelic-fraction-small`, with a horizontal or diagonal bar. `alteration`, `alteration-slash`, and `step` use the text font selected in LilyPond, not the font selected in Ekmelily.

See the example `numeric` which resembles `alteration`.

Custom Notation Styles

A new notation style derived from the currently selected style can be created with the command

```
\ekmUserStyle USERSTYLE #'(
  (ALTERATION ELEMENT ...)
  ...
)
```

USERSTYLE is a freely chosen style name. If it is an empty string `""`, the name of the currently selected style extended with the suffix `-user` is taken.

ALTERATION is a rational number, an alteration name, or the name of a special symbol.

For each specified alteration, a new accidental symbol is defined which is the composition of the ELEMENTs. For all other alterations where the previous accidental symbol of ALTERATION appears likewise (usually combined with other symbols), it is also replaced with the new one. Therefore, the order of alterations in the definition list can be significant, in particular, when a replaced accidental symbol is again defined but for another alteration. See the example `diaQuarter`.

ELEMENT is one of the following:

- a code point (integer), e.g. `#xE47B` for Wilson plus
- a character literal, e.g. `#\b` for flat
- a string, e.g. `"bb"` for double-flat
- markup, e.g. `,(markup #:semisharp)`

Two or more elements are juxtaposed with a padding of 0.12 staff units, but no extra space is inserted between the characters of a string. Note that in a SMuFL compliant font, accidental glyphs usually have a zero side-bearing. This also applies to the Basic Latin (ASCII) characters in the Ekmelos font, so that `"bb"` is drawn without a padding, but `#\b #\b` with a padding. See the example `hewm`.

Special Symbols

Every notation style includes the following special symbols which are assigned to names instead of alterations.

Symbol	Predefined	
`default`	Nothing	Default accidental for alterations without a defined accidental. This applies to the universal notation styles and to some styles for 53 and 96-EDO, or it can occur with `\transpose`.
`leftparen`	`#xE26A`	Left parenthesis for cautionary accidentals.
`rightparen`	`#xE26B`	Right parenthesis for cautionary accidentals.

The special symbols of all predefined styles are equal to the universal style `void`.
They can be customized like accidentals:

```
\ekmUserStyle USERSTYLE #'(
  (default ELEMENT ...)
  (leftparen ELEMENT ...)
  (rightparen ELEMENT ...)
  ...
)
```

If both, the accidental and the parentheses are defined by a single code point, character literal, or string, the entire cautionary accidental is drawn as a single string. This enables the use of a ligature if one is provided by the selected font.

Additional Commands

`\ekmelicOutputSuffix`

Set the name of the selected notation style as the output filename suffix for the current `\book` section.

`\ekmelic-style-name`

Draw the name of the selected notation style as markup.

`\ekmelic-font-name`

Draw the name of the selected font as markup.

`\ekmelic-char ALTERATION`

Draw the accidental of ALTERATION (a rational number or an alteration name) according to the selected notation style as markup.
Used property:

- `font-size` (1)

`\ekmelic-char-text ALTERATION`

Draw the accidental of ALTERATION (a rational number or an alteration name) according to the selected notation style as markup, vertically aligned for use outside of staves, like in `NoteNames`, `ChordNames`, trill spanner, figured bass, or function theory text.
If property `style` is `'chord` the standard accidentals (U+E260 - U+E266) are replaced with the corresponding chord symbols (U+ED60 - U+ED66) and all accidentals are bottom aligned. This is used in `ChordNames`.
Used properties:

- `font-size` (0)
- `style` ('())

`\ekmelic-elem ELEMENT`

Draw ELEMENT with the selected font as markup. ELEMENT is a code point (integer), a character literal, a string, or markup. This command is intended to combine glyphs from the selected font with other markup in a custom notation style.

`\ekm-fraction ARG1 ARG2`

Draw a fraction of ARG1 and ARG2 as markup, with a fraction bar according to property `style`:

- The default `'()` draws a horizontal bar. This is a variant of LilyPond's `\fraction` but with consistent vertical alignment.
- `'slash` draws a diagonal bar.
- `'line` draws a solidus (U+002F) with horizontally stacked numbers.

This command is used by the next two commands.
Used properties:

- `font-size` (0)
- `style` ('())

`\ekmelic-fraction ALTERATION`

Draw ALTERATION (a rational number or an alteration name) as markup, ignoring enharmonical equivalence. If the denominator is 1, only the numerator is drawn, else a fraction according to property `style`, and with `fraction-size` relative to the font size unless `style` is `'line`.
Used properties:

- `fraction-size` (0)
- `style` ('())

`\ekmelic-fraction-small ALTERATION`

Draw ALTERATION (a rational number or an alteration name) as markup like `\ekmelic-fraction` but with a 4 steps smaller fraction size.

`\ekmelic-table NATURAL COMPOSITE ORDER`

Draw a table of all accidentals in the selected notation style as markup, including the natural symbol if NATURAL is true, and all composite accidentals if COMPOSITE is true. Enharmonically equivalent accidentals and special symbols are always ignored. The accidentals are arranged in a row with the respective alteration placed beneath, and sorted by ascending or descending alteration if ORDER is 1 or -1, respectively, or by absolute alteration if ORDER is 2 or -2.
Used properties:

- `font-size` (0)
- `width` (4) : Horizontal extent for each accidental.
- `baseline-skip` : Distance between accidental and alteration.

Examples

Sets the predefined Sims notation style for 72-EDO, selects the English note names, and draws the accidentals with the Bravura font.

```
ekmFont = "Bravura"
\include "ekmel.ily"
\language "english"
\ekmStyle sims
```

Sets the predefined Stein/Zimmermann notation style (`stz`) for quarter-tones (24-EDO) and modifies it into a custom notation style with the Sagittal flat 11 medium diesis down symbol 𝄫 (U+E327) for three-quarter-tones flat, and two sharp symbols 𝄯 (U+E262) for double-sharp.

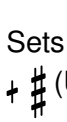
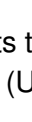
```
\include "ekmel-24.ily"
\ekmStyle stz
\ekmUserStyle myNotation #'(
  (-3/4 #xE327)
  (1 #xE262 #xE262))
```

This is the same example as above but it makes use of the corresponding Scheme symbols.

```
\include "ekmel-24.ily"
\ekmStyle stz
\ekmUserStyle myNotation #`(
  (,THREE-Q-FLAT #xE327)
  (,DOUBLE-SHARP #xE262 #xE262))
```

Sets the HEWM (Helmholtz/Ellis/Wolf/Monzo) notation for 72-EDO and selects the English note names.
Note that double-flat is defined with `#\b #\b` which is drawn with a padding contrary to `"bb"`.

```
\include "ekmel.ily"
\language "english"
\ekmUserStyle hewm #'(
  (1 #\x)
  (-1 #\b #\b)
  (1/2 #\#)
  (-1/2 #\b)
  (1/4 #\^)
  (-1/4 #\v)
  (1/6 #\>)
  (-1/6 #\<)
  (1/12 #\+)
  (-1/12 #\-))
```

Sets the notation after Karlheinz Stockhausen for 24-EDO with the fractional sharp symbols  (U+ED58, U+ED5A), and the quarter-tone flat symbol  (U+ED59).

```
\include "ekmel-24.ily"
\ekmStyle stz
\ekmUserStyle stockhausen #'(
  (1/4 #xED58)
  (-1/4 #xED59)
  (3/4 #xED5A)
  (-3/4 #xED59 #xE260))
```

Sets a variant of the Hesse notation style for 72-EDO using SMuFL characters:
Gould arrows  (U+E27A, U+E27B), Sims half arrows (U+E2A4, U+E2A1), and Bosanquet commatic symbols (U+E479, U+E47A). It is very similar to the Arrow notation style.

```
\include "ekmel.ily"
\ekmUserStyle smuflHesse #'(
  (1/4 #xE27A)
  (-1/4 #xE27B)
  (1/6 #xE2A4)
  (-1/6 #xE2A1)
  (1/12 #xE479)
  (-1/12 #xE47A))
```

Sets the Standard sharp/flat symbols, single thru quintuple, like the predefined Diatonic notation style (`dia`) but for quarter-tones (24-EDO). The quadruple symbols   (U+F61C, U+F61D) are private supplements of Ekmelos. Note that here, the order of alterations is significant since the standard accidentals in the default notation style (`stc`) are rearranged.

```
\include "ekmel-24.ily"
\ekmUserStyle diaQuarter #'(
  (1 #xF61C)
  (-1 #xF61D)
  (3/4 #xE265)
  (-3/4 #xE266)
  (1/2 #xE263)
  (-1/2 #xE264)
  (1/4 #xE262)
  (-1/4 #xE260)
  (5/4 #xF61C #xE262)
  (-5/4 #xF61D #xE260))
```

Sets single glyphs from LilyPond's Emmentaler font for 48-EDO, as well as parentheses for cautionary accidentals. ±9/8 and ±5/4 are omitted here since they are set automatically to combinations of ±1 with ±1/8 and ±1/4.

```
\include "ekmel-48.ily"
\ekmUserStyle lilysingle #`(
  (0    ,(markup #:natural))
  (1/8  ,(markup #:musicglyph "accidentals.natural.arrowup"))
  (-1/8 ,(markup #:musicglyph "accidentals.natural.arrowdown"))
  (1/4  ,(markup #:semisharp))
  (-1/4 ,(markup #:semiflat))
  (3/8  ,(markup #:musicglyph "accidentals.sharp.arrowdown"))
  (-3/8 ,(markup #:musicglyph "accidentals.flat.arrowup"))
  (1/2  ,(markup #:sharp))
  (-1/2 ,(markup #:flat))
  (5/8  ,(markup #:musicglyph "accidentals.sharp.arrowup"))
  (-5/8 ,(markup #:musicglyph "accidentals.flat.arrowdown"))
  (3/4  ,(markup #:sesquisharp))
  (-3/4 ,(markup #:sesquiflat))
  (7/8  ,(markup #:musicglyph "accidentals.sharp.slashslashslash.stemstem"))
  (-7/8 ,(markup #:musicglyph "accidentals.flatflat.slash"))
  (1    ,(markup #:doublesharp))
  (-1   ,(markup #:doubleflat))
  (leftparen ,(markup #:musicglyph "accidentals.leftparen"))
  (rightparen ,(markup #:musicglyph "accidentals.rightparen")))
```

Sets alteration values instead of accidental symbols with the default accidental in the void notation style. The values are drawn with the command `\ekmelic-fraction` and with a 3 steps smaller font size. It is similar to the alteration notation style.

```
#(define-markup-command (numeric-accidental layout props)
  ()
  (let ((alt (ly:chain-assoc-get 'alteration props 0)))
    (interpret-markup layout
      (cons '((font-size . -3)) props)
      (markup #:vcenter #:ekmelic-fraction alt))))

\ekmStyle void
\ekmUserStyle numeric #`(
  (default ,(markup #:numeric-accidental)))
```

See also `user-styles.ly` .