

LilyPond

The music typesetter

Changes

The LilyPond development team

This document lists changes and new features in LilyPond version 2.25.4 since 2.24.

For more information about how this manual fits with the other documentation, or to read this manual in other formats, see Section “Manuals” in *General Information*.

If you are missing any manuals, the complete documentation can be found at <https://lilypond.org/>.

This document has been placed in the public domain.

For LilyPond version 2.25.4

Note: LilyPond releases can contain syntax changes, which may require modifications in your existing files written for older versions so that they work in the new version. To upgrade files, it is **strongly recommended** to use the `convert-ly` tool distributed with LilyPond, which is described in Section “Updating files with `convert-ly`” in *Application Usage*. `convert-ly` can perform almost all syntax updates automatically. Frescobaldi users can run `convert-ly` directly from Frescobaldi using “Tools > Update with `convert-ly`...”. Other editing environments with LilyPond support may provide a way to run `convert-ly` graphically.

Major changes in LilyPond

- Margins are now wider by default following the general layout of several publishers (and the recommendations of Elaine Gould).

In order to switch back to the previous settings (e.g., to keep the same layout when upgrading an existing score to version 2.25.4), add the following code:

```
\paper {
  top-margin = 5\mm
  bottom-margin = 10\mm
  top-system-spacing.basic-distance = 1
  top-markup-spacing.basic-distance = 0
  left-margin = 10\mm
  right-margin = 10\mm
  inner-margin = 10\mm
  outer-margin = 20\mm
  binding-offset = 0\mm
}
```

- Instead of generating PostScript or SVG output by itself, LilyPond can now use the Cairo library to produce its output. This is referred to as the ‘Cairo backend’, and can be turned on using the `-dbackend=cairo` command-line option. This works for all output formats (PDF, SVG, PNG, PostScript), and brings speed and rendering fidelity improvements in SVG output in particular. However, keep in mind that this backend does not yet implement all features of the default backends. Among the features not currently supported are PDF outlines, the `-dembed-source-code` option for PDF, and the `output-attributes` property for SVG.

New for musical notation

Pitches improvements

None so far.

Rhythm improvements

- The new option `span-all-note-heads` may be used to make tuplet brackets span all note heads (not just the stems) as recommended by Gould and Ross.



Expressive mark improvements

- Two new variant glyphs for breathing signs are available: `'laltcomma'` and `'raltcomma'`. These glyphs represent the old shapes of `'lcomma'` and `'rcomma'`, respectively, before changing them to more common shapes.

```
{
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.raltcomma" }
  f'2 \breathe f' |
}
```



Repeat improvements

- `\repeat volta` alternative endings no longer create invisible bar lines. This may affect line breaking, horizontal spacing, and `VoltaBracket` extent where an alternative begins or ends without a bar line. In the case of an undesired change, try adding `\bar ""` or another command that creates a `BarLine` at that point.
- Using the new `printInitialRepeatBar` property, it is possible to make a start repeat bar line automatically printed even at the beginning of the piece.



Editorial annotation improvements

- `NoteName` grobs are now horizontally centered by default.

Text and font improvements

- The syntax for customizing text and music fonts has been changed. Instead of

```
\paper {
  #(define fonts
    (set-global-fonts
```

```

#:music "Name of music font"
#:brace "Name of music brace font"
#:roman "Name of roman font"
#:sans "Name of sans-serif font"
#:typewriter "Name of typewriter font"))
}

```

or

```

\paper {
  #(define fonts
    (make-pango-font-tree
      "Name of roman font"
      "Name of sans-serif font"
      "Name of typewriter font"
      factor))
}

```

the new syntax is

```

\paper {
  fonts.music = "Name of music font"
  fonts.roman = "Name of roman font"
  fonts.sans = "Name of sans-serif font"
  fonts.typewriter = "Name of typewriter font"
}

```

Unlike the previous syntax, the new syntax does not interfere with font sizes, which should be set separately using `set-global-staff-size` or `layout-set-staff-size`.

There is no brace key in the fonts alist; braces glyphs now always default to the music font. However, it is still possible to override this by using an extra font family, as shown in this example (which requires the LilyJAZZ font):

```

\paper {
  fonts.music-alt = "lilyjazz"
}

\layout {
  \context {
    \Score
    \override SystemStartBrace.font-family = #'music-alt
  }
}

\new PianoStaff <<
  \new Staff { c' }
  \new Staff { c' }
>>

\markup \override #'(font-family . music-alt) \left-brace #20

```

- The `\lookup` markup command can now only be used for braces; for other glyphs, use the `\musicglyph` command. Instead of `\lookup`, it is also generally recommended to use `\left-brace`.
- In markup, when a music font is used (such as for dynamic markings), a glyph absent from the music font was previously rendered in a normal text font. This is no longer the case;

a warning about the missing glyph is output instead. In order to use a text font, use the `\text` markup command. For example:

```
dolceP =
#(make-dynamic-script
#{
  \markup {
    \text \normal-weight dolce
    p
  }
#})

{ c'\dolceP }
```



- Small caps are now achieved by overriding `font-variant` to `small-caps` instead of overriding `font-shape` to `caps`. Since `font-shape` is primarily for achieving italics, this change makes it possible to use small caps and italics at the same time.

Small caps in italics

- The `font-series` property is now more flexible and allows to specify values such as `semibold` and `light` instead of only `normal` and `bold`.

The `medium` value is now an intermediate value between `normal` and `bold` rather than an equivalent of `normal`. Accordingly, the `\medium` markup command has been renamed to `\normal-weight`.

- The new `font-stretch` property allows selecting a condensed or expanded font.
- The text of a `VoltaBracket` grob, as set by `\override Score.VoltaBracket.text = ...` or `\set Score.repeatCommands = ...`, is no longer automatically typeset in a music font; use the `\volta-number` markup command on those parts that need to be. For example, convert

```
\set Score.repeatCommands = #'((volta "2, 5"))
```

to

```
\set Score.repeatCommands =
#`((volta ,#{ \markup {
  \concat { \volta-number 2 , }
  \volta-number 5 }
#}))
```

New for specialist notation

- For orthogonality with other ancient clefs, five new mensural clefs are available: "mensural-f2", "mensural-f3", "mensural-f4" (same as "mensural-f"), "mensural-f5", "mensural-g1", "mensural-g2" (same as "mensural-g").
- The default time signature and accidental style in a PetrucciStaff context is now the same as in MensuralStaff.
- White mensural ligatures now support some rare ligatures (semibreves alone or in the middle), and allow tweaks to show some non-necessary stems.

```

\score {
  \relative {
    \set Score.timing = ##f
    \set Score.measureBarType = #'()
    \override NoteHead.style = #'petrucci
    \override Staff.TimeSignature.style = #'mensural
    \clef "petrucci-c4"
    \[ a1 g f e \]
    \[ a1 g\longa \]
    \[ \once \override NoteHead.left-down-stem = ##t
      a\breve b
      \once \override NoteHead.right-down-stem = ##t
      g\longa \]
    \[ \once \override NoteHead.right-down-stem = ##t
      b\maxima
      \once \override NoteHead.right-up-stem = ##t
      g\longa \]
  }
  \layout {
    \context {
      \Voice
      \remove Ligature_bracket_engraver
      \consists Mensural_ligature_engraver
    }
  }
}

```



Miscellaneous improvements

- Embedding PNG images is now supported using the new `\image` markup command. This supplements the existing `\epsfile` command for EPS images.
`\image` works for both PNG and EPS images. For EPS images, the difference between using `\image` and `\epsfile` is that `\image` adds a white background by default, while `\epsfile` does not.
- The new `\qr-code` markup command inserts a QR code of the specified size for the specified URL. This can be used to link to, e.g., the website of a composer or publisher, the LilyPond source files for the score, recordings, etc.

```
\markup \qr-code #10 "https://lilypond.org"
```

