LilyPond

Changes

The LilyPond development team

This document lists changes and new features in LilyPond version 2.23.10 since 2.22.

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 2.23.10
• In figured bass, _ now creates an empty figure that still takes up space.

```
\figures{<8 _ 4> <_ 5+ 3>
}
```

![Figured Bass Example]

• Support for chord grids has been added.

```
G^\Delta
Bb m^7
\ respectful
\ regards
Am^7
B^7#11
\end{example}
```

• Formatting of figured bass has been improved. In particular, the default size is reduced to a value used by many Urtext editions of Baroque music.

• Text variant glyphs for sharp, flat, natural, double sharp, and double flat are now available in the Emmentaler fonts. In markup, they can be easily accessed with standard Unicode values.

```
```

• \defineBarLine now accepts #t in lieu of repeating the mid-line glyph name.

• The new markup command \align-on-other translates a markup as if it was aligned to another markup.

```
1
12
12345
123
```

• It is now possible to control the width and the shape of (some) Emmentaler digits using OpenType features.

```
0123456789 (time signatures)
```

```
0123456789 (fingering)
```

```
0123456789 (fixed-width digits)
```

• The bar line type "-" has been removed. convert-ly converts it to ".". There is a slight difference in horizontal spacing at line breaks.

• automaticBars has been removed. convert-ly converts automaticBars = ##f to measureBarType = '#()'.

![Chord Grid Example]
• The \break command now always inserts a break, bypassing all default decisions about break points. For example, it is no longer necessary to insert \bar "" to obtain a mid-measure break.

The new \allowBreak command inserts a possible break point, without forcing it, but bypassing default decisions like \break does.

• In ChordNames, multi-measure rests now also cause the “N.C.” symbol to be printed, just like normal rests.

• Scheme translators can now contain listeners written as

   (listeners
    ((event-class engraver event #:once)
     ...
    ))

These are never triggered more than once per time step. They emit a warning if they receive two events in the same time step, except if the events are equal.

• The default appearance of trill spanners has changed to better match classical engraving conventions. They now end before the next note, not over it. If the next note has an accidental, they stop before it. If the next note is the first note of a measure, they stop over the bar line instead.

• New markup commands \with-true-dimension and \with-true-dimensions are available. They give the markup the actual extent(s) of its printed ink, which may differ from the default extents for some font glyphs due to text regularity constraints.

• The new context property breathMarkType selects the mark that \breathe produces from several predefined types.

   \fixed c' \{ 
    \set breathMarkType = #'tickmark 
    c2 \breathe d2 
   \}

• Properties of PaperColumn and NonMusicalPaperColumn (such as NonMusicalPaperColumn.line-break-system-details) can now be overridden mid-music with the usual command \once \override. They used to be a special case requiring the \overrideProperty command.

• Scheme translators can now define a new slot called pre-process-music. It is called on all translators, after all listeners but before all process-music slots. This can be used for processing that depends on all events heard but needs to set context properties before other translators read them.

• \markup \path now also works in SVG output even if the path does not begin with a moveto or rmoveto command. Also, it now accepts single-letter SVG equivalents (moveto = M, etc.).
• A new context VaticanaLyrics is available. It is similar to Lyrics, providing a hyphenation style (a single, flush-left hyphen between two syllables) as used in the notational style of Editio Vaticana.

• Two redundant glyphs in the Emmentaler font have been removed: scripts.trillelement (use scripts.trill_element instead) and scripts.augmentum (use dots.dotvaticana instead).

• Alterations in figured bass are now printed with a larger size for better readability; their vertical positioning is improved, too.

• Two new markup functions \with-dimension and \with-dimension-from are available. They are similar to \with-dimensions and \with-dimensions-from, respectively, modifying only a single dimension (instead of both).

• The \enablePolymeter command is now provided as an input shorthand for moving engravers as is necessary to allow different time signatures in parallel. The code:

\layout {
  \context {
    \Score
    \remove Timing_translator
    \remove Default_bar_line_engraver
  }
  \context {
    \Staff
    \consists Timing_translator
    \consists Default_bar_line_engraver
  }
}

can thus be shortened as:

\layout {
  \enablePolymeter
}

Independent of this, Default_bar_line_engraver has been removed.

• \bar "" is no longer necessary to print the first bar number. It now suffices to set barNumberVisibility to all-bar-numbers-visible, or one of the other visibility settings where the first bar number is visible.

Note that this is a change in behavior for scores that set barNumberVisibility to all-bar-numbers-visible or such and BarNumber.break-visibility to #t without having \bar "". Now, a bar number is printed at the beginning. This is just the expected behavior (all bar numbers should be visible), but due to slightly unclear documentation, users may have used these settings to print bar numbers in the middle of systems except for the first bar number. In such cases, simply remove \set Score.barNumberVisibility = #all-bar-numbers-visible since \override BarNumber.break-visibility = ##t does the relevant setting alone.

• The \markup->string function converts a markup into an approximate string representation; it is used for outputting PDF metadata as well as MIDI lyrics and markers. Markup commands can now define a custom method to convert markups created using them into strings, for use by \markup->string. For example:

  #(define-markup-command (upcase layout props arg) (string?)
    #:as-string (string-upcase arg)
    (interpret-markup layout props (string-upcase arg)))
• \bar "," creates a short bar line.

\begin{music}
\bar c \quad \bar c
\end{music}

• The new show-horizontal-skylines and show-vertical-skylines properties allow to display an object’s skylines. This is more flexible than the already existing debug-skylines option because it works for all grobs. While primarily meant for debugging LilyPond, this can be useful when trying to understand spacing decisions or overriding stencils in Scheme.

\begin{music}
\clef \guitar
c \quad \sharp
\end{music}

• Balloons now have changeable formatting.

\begin{music}
\bar c \quad \bar c
\end{music}

• The new \repeat segno command automatically notates a variety of da-capo and dal-segno forms.

\begin{music}
\repeat segno 2 \{b1\} \fine
\end{music}

\begin{music}
\score { \music }
\score { \unfoldRepeats \music }
\end{music}

\begin{music}
D.C.
\end{music}

• The new \fine command inserts a final bar line that interacts gracefully with repeat bar lines. Used inside \repeat, it also prints Fine and ends the music after unfolding.

\begin{music}
\repeat volta 2 \{f1 \volta 2 \fine b1\}
\end{music}
The new \section command inserts a double bar line that interacts gracefully with repeat bar lines. A passage can be named with the new \sectionLabel command.

\fixed c' {
  f1
  \break
  \section
  \sectionLabel "Trio"
  \repeat volta 2 {
    b1
  }
}

The drum notation style \add{weinberg-drums-style} was added. It is based on Norman Weinger's standardization work.

Support for Persian classical music is now available. For this, two accidental glyphs, sor\i and kor\o\o\n, have been added to LilyPond.

\include "persian.ly"

\relative c' {
  \key d \chahargah
  bk'8 a gs fo r g ak g |
  fs ek d c d ef16 d c4 |
}

The new \add{Mark_performer} creates MIDI Marker events like \add{Mark_engraver} creates printed marks.
• The new Mark_tracking_translator takes over from Mark_engraver the decision of when to create a mark. Mark_engraver continues to control formatting and vertical placement. By default, Mark_engravers in multiple contexts create a common sequence of marks. If independent sequences are desired, multiple Mark_tracking_translators must be used.

• The default of the aux-files program option changed to #f. If you are calling LilyPond with the -dbackend=eps argument and need the auxiliary .tex and .texi files, you now have to specify -daux-files explicitly. The formats for lilypond-book images can be set separately for the tall page image (typically PNG for HTML output) and per-system images (typically, EPS or PDF for printed output) with the -d sub-options -dtall-page-formats and -dseparate-page-formats respectively.

• Events attached to notes (e.g., dynamics or articulations) can be delayed by an arbitrary duration using \after. This simplifies many situations that previously required the use of explicit polyphony and spacer rests.

\{
\after 2 \turn g'2. a'4
\after 2 \< b'1
\after 2. \f c''
<\< \after 4 \> \after 2\! d''
\}

\begin{music}
\bar{\bar{f}} \infty \bar{\bar{f}} \bar{\bar{f}} \bar{\bar{f}} \bar{\bar{f}} \bar{\bar{f}} \bar{\bar{f}}\end{music}

• lilypond-book supports two new music fragment options paper-width and paper-height to set a custom paper size.

• set-default-paper-size and set-paper-size now accept a custom paper size.

#(set-default-paper-size '(cons (* 100 mm) (* 50 mm)))

• Scheme-defined translators usable in both \layout and \midi can now be created with make-translator. Scheme-defined performers usable only in \midi can now be created with make-performer. Those macros work strictly like the previously existing macro make-engraver for creating engravers only usable in \layout.

• The syntax for conditions in markups was made more flexible and user-friendly. It uses the new markup commands \if and \unless. Here are example replacements:

\begin{verbatim}
2.22 syntax  \if 2.24 syntax  \unless
\on-the-fly #first-page ... \if \on-first-page ...
\on-the-fly #not-part-first-page ... \unless \on-first-page-of-part ...
\on-the-fly #(on-page n) ... \if \on-page #n ...
\end{verbatim}

• In figured bass, brackets can now also be added around accidentals.

\begin{music}
\accidentals.natural.figbass/figbass.five
\accidentals.sharp.figbass/figbass.three/figbass.five/figbass.seven
\end{music}

• \numericTimeSignature and \defaultTimeSignature now act on all staves at once (more precisely, on all staves in the same Timing context), thus matching the behavior of \time.

• Parenthesizing chords is supported. Currently, the font size of the parentheses has to be adjusted manually.
- Parenthesizing spanners is supported.

- A “time-based” version of the \parenthesize command was added. It takes a grob path: \parenthesize GrobName or \parenthesize ContextName.GrobName. It acts like a \once \override. This interface complements the already existing form \parenthesize event, in a fashion similar to \footnote.

{  
\parenthesize NoteHead  
c'1  
\parenthesize Staff.KeySignature  
\key g \major  
c'1  
}

- The same grob definition can now be used to create grobs of different classes (Item, Spanner, Paper_column, System). As part of this change, the grob types FootnoteItem and FootnoteSpanner were consolidated into a single type Footnote. Similarly, BalloonTextSpanner and BalloonTextItem are unified into BalloonText.

When the grob definition does not mandate a class, engravers should choose what class to create a grob with. For authors of Scheme engravers, this means using either ly:engraver-make-item or ly:engraver-make-spanner. The utility function ly:engraver-make-sticky is provided to support the frequent case of sticky grobs, such as footnotes and balloons. It creates a grob with the same class as another grob and administrates parents and bounds.

- The new command \vshape is like \shape, but also shows the control points and polygon for easier tweaking.

{ a1\vshape '#'((0 . 0) (0 . 0.5) (0 . 0.9) (0 . 0.4))~( c'1) }

- Support for alternate accidentals was improved. Through the alterationGlyphs property of staff-like contexts, accidental glyphs may be set for all grobs at once (refer to Section “Alternate accidental glyphs” in Notation Reference).
• Ottava brackets may apply to a single voice instead of the entire staff. This used to require workarounds.

```latex
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}
```

• Measure counts now take compressed multi-measure rests and alternatives into account.

```latex
\layout {
  \context {
    \Score
    \centerBarNumbers = ##t
  }
}
```

• The string tunings banjo-double-c and banjo-double-d were added.

• lilypond-book now appends the current directory as the last entry to search for included files, instead of prepending it to the list of specified include paths. This allows include directories to shadow files from the current directory, and will only be noticed if there are files with the same name in both.

• lilypond-book supports a new snippet option inline for inline music, that is, music snippets like that appear within a paragraph of text.

• Bar numbers may be centered in their measure, as is common in film scores.
Adding the Melody_engraver to the Voice context now works out of the box to change the stem direction of the middle note according to the melody. Previously, this required an additional override to Stem.neutral-direction.

\new Voice \with {
  \consists Melody_engraver
}
\relative c'' {
  \autoBeamOff
  g8 b a e g b a g |
  c b d c b e d c |
}

In the Emmentaler font, identical-looking noteheads whose only difference was stem direction have been consolidated into a single glyph. For instance, the glyphs noteheads.u2triangle and noteheads.d2triangle have been replaced by a single glyph, noteheads.s2triangle. Notehead pairs that look visually different depending on the direction remain distinct.

In addition, the stem-attachment property of NoteHead grobs now returns its actual, direction-dependent stem attachment point instead of a hypothetical upwards-stem attachment point.

Repeat alternatives may appear within the repeated section.

\repeat volta 3 { c'1 \alternative { d' e' } f' }

The volta numbers for repeat alternatives may be set with the \volta command.

\repeat volta 3 c'1 \alternative { \volta 1 d' \volta 2,3 e' }
• The \texttt{\textbackslash volta} command removes music when a repeat is unfolded.
• The \texttt{\textbackslash unfolded} command adds music when a repeat is unfolded.
• The ends of hairpins may now be aligned to the LEFT, CENTER or RIGHT of NoteColumn grobs by overriding the property endpoint-alignments.

• The \texttt{lilypond-book} script now allows braces in the argument of the commands \texttt{\textbackslash lilypond} (for LaTeX) and \texttt{@lilypond} (for Texinfo).
• With the new markup list command \texttt{string-lines} it is now possible to split a string at a given character. The default is to split at line break. Surrounding white space gets dropped. The resulting list of markups may be further formatted. This is a very convenient way of entering additional stanzas for songs.

Twinkle, twinkle, little star,  
How I wonder what you are!  
Up above the world so high,  
Like a diamond in the sky.  
Twinkle, twinkle, little star,  
How I wonder what you are!

• A new grob \texttt{FingerGlideSpanner} is now available, indicating a finger gliding on a string from one to another position. Several appearances are possible, depending on the setting of style. Shown in the image are line, stub-left, stub-right and stub-both.

Also possible is dashed-line, dotted-line, zigzag, trill, bow and none.

• A new grob \texttt{BendSpanner} is now available for TabStaff, indicating a bent string. Apart from the default three styles are possible: ’hold, ’pre-bend and ’pre-bend-hold.

• A new grob callback function \texttt{break-alignment-list} is now available for returning different values depending on a grob’s break direction. As an example, use it to provide different
alignments of a grob depending on whether it is positioned at the beginning, the middle, or the end of a line.

- The alignment of bar numbers appearing in the middle or end of a system has been changed to align them on their left edge. This is in keeping with the advice of Elaine Gould (*Behind Bars*, p. 237), and was mostly the consensus reached in a discussion of the issue by developers. The alignment of bars at the beginning of a system remains unchanged.