LilyPond

The music typesetter

Changes

The LilyPond development team

> This document lists changes and new features in LilyPond version 2.23.5 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 `http://lilypond.org/`.

For LilyPond version 2.23.5

- 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''
}
```
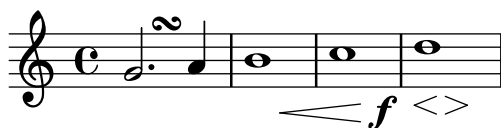

- `lilypond-book` supports two new music fragment options `paper-width` and `paper-height` to set a custom paper size.

- `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:

2.22 syntax	2.24 syntax
\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 ...

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

- `\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.
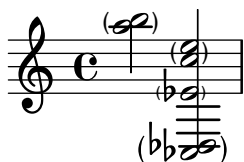
- Parenthesizing spanners is supported.

- 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.

```
\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.

- 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.

```
\layout {
  \context {
    \Score
    centerBarNumbers = ##t
    barNumberVisibility = #all-bar-numbers-visible
  }
}

<<
```

```
    { \bar "" \repeat unfold 3 { c'4 d' e' f' } }
    { \repeat unfold 3 { c'4 d' e' f' } }
  >>
```


- 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 |
}
```


  The `suspendMelodyDecisions` context property may be used to turn off this behavior temporarily, as `\override Stem.neutral-direction = #DOWN` used to do.

- 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 in places other than the end of 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 `\volta` command removes music when a repeat is unfolded.

- The `\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 `lilypond-book` script now allows braces in the argument of the commands `\lilypond` (for LaTeX) and `@lilypond` (for Texinfo).
- With the new markup list command `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!

- Color transparency is supported when generating Postscript and PDF files. To use this new feature, it is necessary to use Ghostscript version 9.53 or later.
- A new grob `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`.
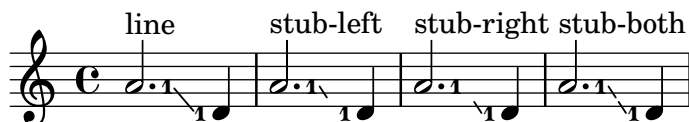
Also possible is `dashed-line`, `dotted-line`, `zigzag`, `trill`, `bow` and `none`.

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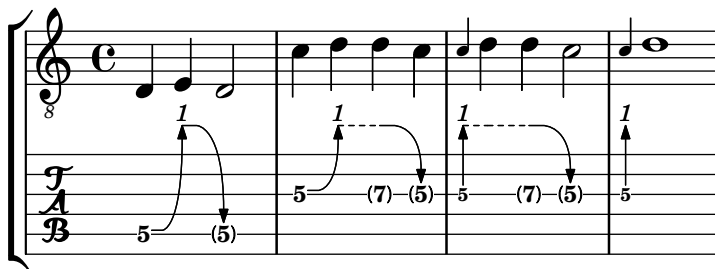
- A new grob callback function `break-alignment-list` is now available for returning different values depending on a grob's break direction. As an example, use it to provide different alignments of a grob depending on whether it is positioned at the beginning, the middle, or the end of a line.

- 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.