LilyPond

The music typesetter

Changes

The LilyPond development team

This document lists changes and new features in LilyPond version 2.21.82 since 2.20.

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/`.
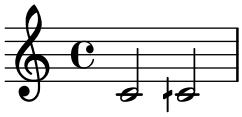
This document has been placed in the public domain.

For LilyPond version 2.21.82

New for musical notation
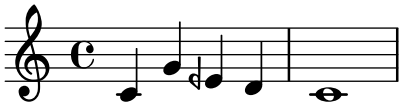
Displaying pitch improvements

- An accidental glyph corresponding to SMuFL code U+E2E3 has been added (Extended Helmholtz-Ellis accidentals: Raise by one undecimal quartertone).
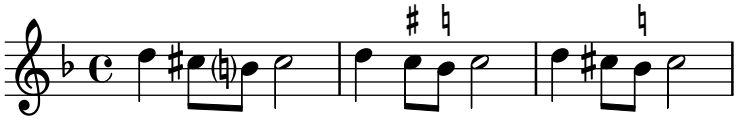


- Quarter-tone note names are now available in all the input languages.

```
\language "català"
\relative { do'4 sol' miqb re do1 }
```
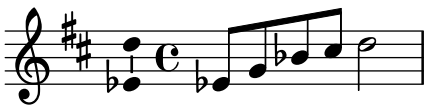


- Setting `suggestAccidentals` to `'cautionary` turns only cautionary accidentals into `AccidentalSuggestion`s. This can be used to distinguish between facsimile and editorial accidentals.

```
\relative {
  \key d \minor
  d''4 cis8 b? cis2
  \set suggestAccidentals = ##t
  d4 cis8 b? cis2
  \set suggestAccidentals = #'cautionary
  d4 cis8 b? cis2
}
```



- The command `\ambitusAfter` has been added. It is used to move the ambitus to a different position.
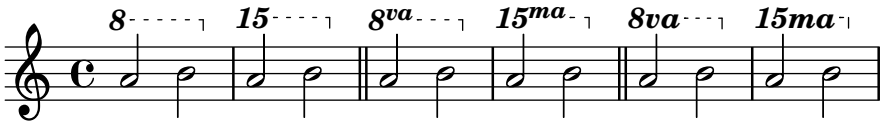
```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



- Ottava brackets are now introduced with a single number by default (e.g., ‘8’ or ‘15’), and printed in bold. A new `ottavationMarkups` property has been introduced to modify that behavior, with several predefined lists of markups available:

```
\relative c'' {
```
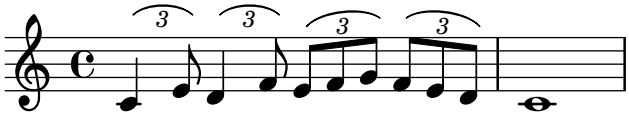
```
  \ottava #1
  a'2 b
  \ottava #2
  a'2 b
  \bar "||"
  \set Staff.ottavationMarkups = #ottavation-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
  \bar "||"
  \set Staff.ottavationMarkups = #ottavation-simple-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
}
```



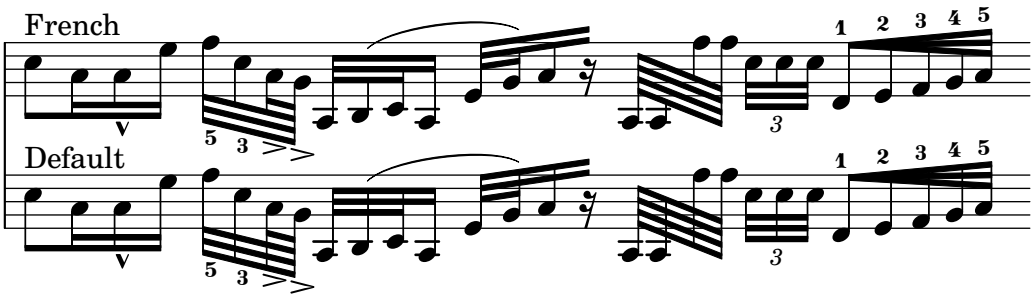
Rhythm improvements

- Tuplets can now be printed with slurs instead of brackets:

```
\relative {
  \tuplet 3/2 4 {
    \override TupletBracket.tuplet-slur = ##t
    c'4 e8 d4 f8
    \override TupletBracket.bracket-visibility = ##t
    e f g f e d
  } c1
}
```

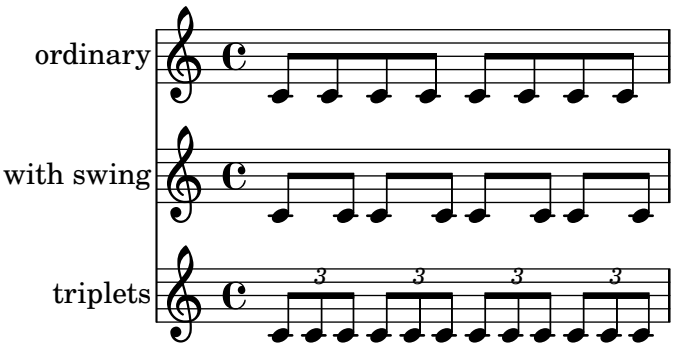


- French beaming (`\override Stem.french-beaming = ##t`) now *exactly* behaves like standard (i.e., default) beaming in every respect (beam positioning and placement of any articulation, fingering, etc.). The only remaining difference are inner stems not passing through beams.

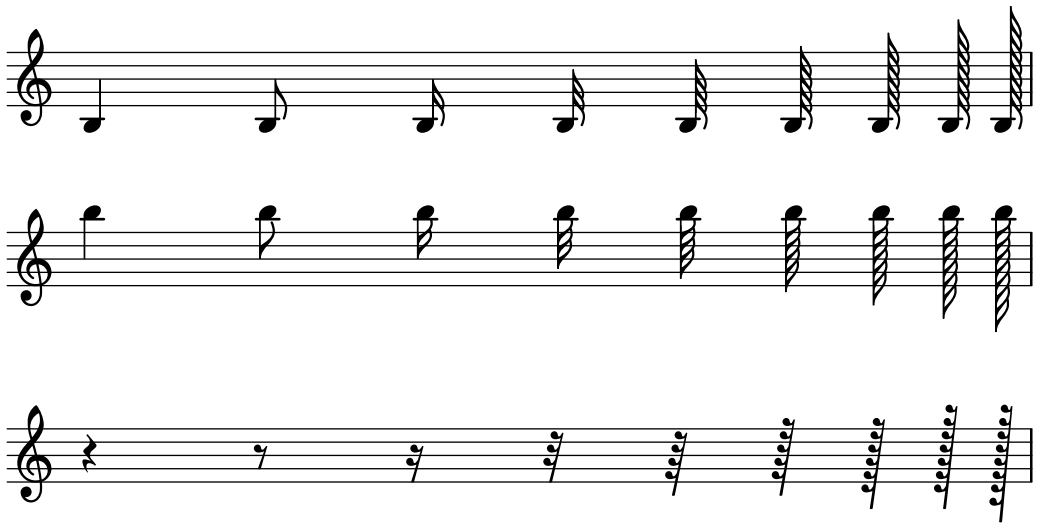


- Swing and irregular rhythmic patterns may now be applied to music expressions made of regular durations, which may be used to render inequal rhythmic interpretation in MIDI.

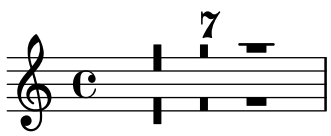
```
\include "swing.ly"
<<
  \new Staff \with {
    instrumentName = "ordinary"
  } \repeat unfold 8 c'8
  \new Staff \with {
    instrumentName = "with swing"
  } \repeat unfold 4 \tripletFeel 8 { c'8 c' }
  \new Staff \with {
    instrumentName = "triplets"
  } \tuplet 3/2 4 \repeat unfold 12 c'8
>>
```



- Font glyphs for 256th, 512th, and 1024th flags and rests have been added.

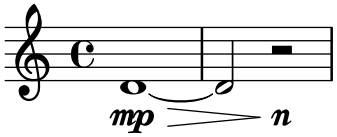


- The new `Merge_mmrest_numbers_engraver` hides duplicate numbers on multi-measure rests.
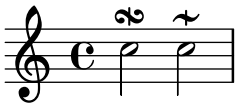


Expressive mark improvements

- A dynamic command `\n` for *niente* has been added.



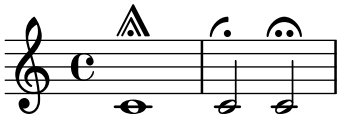
- Two new ornaments have been added.

```
{
  c''2\slashturn c''\haydnturn
}
```



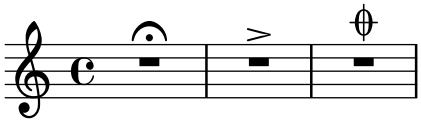
- A very short fermata and Henze variants of long and short fermatas have been added.

```
{
  c'1\veryshortfermata
  c'2\henzeshortfermata c'\henzelongfermata
}
```



- Fermatas and other articulations can now be added directly to multi-measure rests. Therefore the command `\fermataMarkup` is deprecated.

```
{
  R1\fermata
  R->
  R\coda
}
```

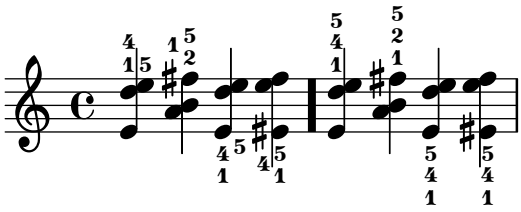


Editorial annotation improvements

- The `New_fingering_engraver` will now consider the `X-align-on-main-noteheads` property of the `self-alignment-interface`. If set to true (`##t`), all fingerings oriented `up` or `down` will be arranged in a straight column and aligned on the noteheads on the correct side of the stem:

```
{
  \set fingeringOrientations = #'(up)
  <e'-1 d''-4 e''-5>4 <a'-1 b'-2 fis''-5>
  \set fingeringOrientations = #'(down)
  <e'-1 d''-4 e''-5> <eis'-1 e''-4 fis''-5>
  \bar "."
  \override Fingering.X-align-on-main-noteheads = ##t
```

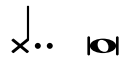
```
  \set fingeringOrientations = #'(up)
  <e'-1 d''-4 e''-5> <a'-1 b'-2 fis''-5>
  \set fingeringOrientations = #'(down)
  <e'-1 d''-4 e''-5> <eis'-1 e''-4 fis''-5>
}
```



Text formatting improvements

- The default fonts are now those from the URW Core 35 set, version 2.0, replacing the TeX Gyre fonts. Note that the fonts come with a ligature for 'Nr.'; see Section "Fonts explained" in *Notation Reference* for solutions to avoid it locally and globally.
- The `\note` markup command now takes as its first argument a duration instead of a string:

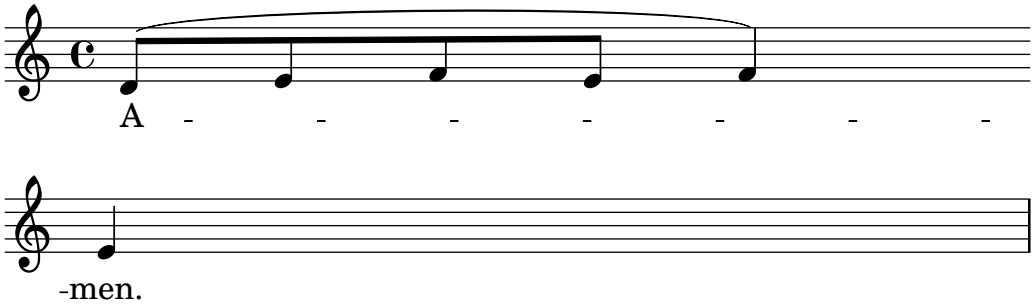
```
\markup {
  \override #'(style . cross) {
    \note {4..} #UP
  }
  \hspace #2
  \note {\breve} #0
}
```



New for specialist notation
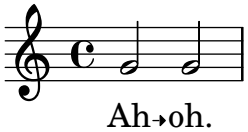
Vocal music improvements

- A lyric hyphen may now be repeated at the start of a system beginning with a new syllable:



- A gradual change of vowel (or sustained consonant) may be indicated by adding a vowel transition between lyric syllables with the command `\vowelTransition`.

```
{ g'2 g' }
\addlyrics { Ah \vowelTransition oh. }
```



Unfretted and fretted string instrument improvements

- Fret-diagrams may now be printed left-handed, setting `handedness` to `LEFT` (as a subproperty of `fret-diagram-details`).

```
\markup
  \center-column {
    "C"
    "(left-handed)"
     \override #`(fret-diagram-details . ((handedness . ,LEFT)))
     \fret-diagram "6-x;5-3-3;4-2-2;3-o;2-1;1-o;"
  }
```

C
(left-handed)

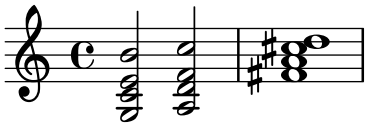


- Some ukulele fretboard-diagrams have been fixed, and new diagrams have also been added.

Chord notation improvements

- Chords may now be automatically inverted or voiced with dropped notes.
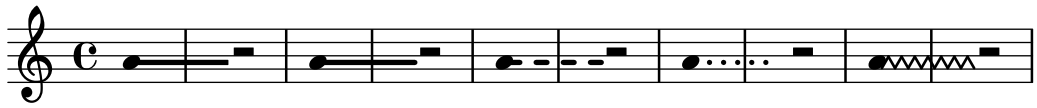
```
\chordmode {
  \dropNote 2 {
    c2:maj7 d:m7
  }
  \invertChords 1 d1:maj7
}
```

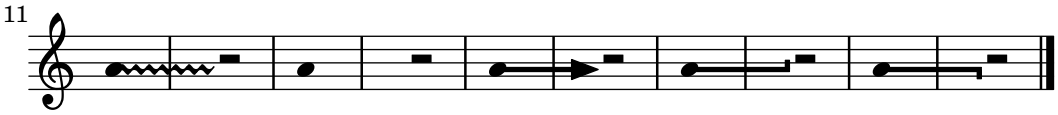


- Legacy chord naming functions `banter-chord-names` and `jazz-chord-names`, have been removed from the main codebase, as have been the properties `chordNamesExceptionsFull` and `chordNamesExceptionsPartial`. As part of that rewrite, power chords are now included in the default exceptions; they will be printed correctly (with a superscript like all other chords) with no additional tweaks required, thereby removing the need for the `\powerChords` predefined command. Anyone interested in the legacy functions may find a (somewhat working) copy of them in the `chord-names-alternative.ly` snippet.

Contemporary music improvements

- A new grob `DurationLine` is now available. It continues a rhythmic grob with a line, ending at the next rhythmic grob. Possible styles are `'beam`, `'line`, `'dashed-line`, `'dotted-line`, `'zigzag`, `'trill` and `'none`. The duration line may end with a hook (beam-style only) or an arrow.





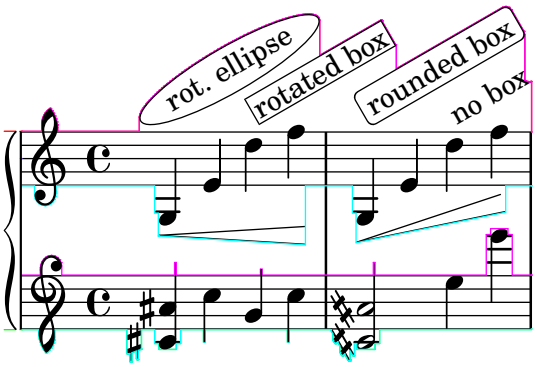
New for input and output

Input file improvements

- As announced in version 2.17.3 nearly eight years ago, the `'relative-includes` option is now enabled by default; included files that contain an `\include` command of their own must account for their own path rather than the main file's directory. That behavior may however be switched off by setting `'relative-includes` to `#f`, either as a command line option or using `ly:set-option` in source files.
- `\compressFullBarRests` has been renamed to `\compressEmptyMeasures`, to avoid possible confusion with `\compressMMRests`. Likewise, `\expandFullBarRests` has now become `\expandEmptyMeasures`.
- The `\partcombine` command, as well as all `partCombine`-prefixed commands, subroutines and property names, are now written with a capital C, such as `\partCombine`, `\partCombineApart` etc. The `\autochange` command is now also capitalized as `\autoChange`.
- All input languages (`\language` statement) can be entered using their proper UTF-8 spelling (i.e., including special characters). The missing names `català` and `português` are available now in addition to the original names `catalan` and `portugues`.
- LilyPond for Windows (MinGW) can handle Unicode filenames on Windows 10 1903 and above.

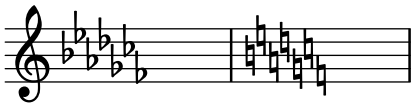
Output improvements

- Skylines will now take account of the `rotation` property of layout objects. Rotating a crescendo hairpin by applying `\override Hairpin.rotation = #'(15 0 0)`, for instance, will now actually have influence on the skylines and thus help to enhance spacing.



- Slight padding added between natural glyphs just touching at the corners in key cancellations.

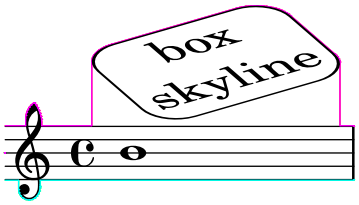
```
{
  \omit Staff.TimeSignature
  \key ces \major s1 \key c \major s
}
```



- Skylines of boxes now reflect the actual box outline including rounded corners and rotation.
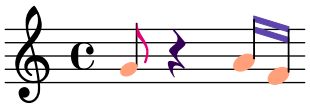
```
#(ly:set-option 'debug-skylines #t)
```

```
{
  b'1^\markup \scale #'(2 . 1)
              \rotate #30
              \override #'(corner-radius . 3)
              \rounded-box \center-column { box skyline }
}
```



- CSS-style colors can now be used directly as text strings; either with predefined color names (like with the already available `css-color` function), or with hexadecimal color codes prefixed with `#`. All stencils that support a `color` property now accept either a list or a string; in the latter case, that string is used directly in the SVG output. This allows to use alpha transparency (entered as "#RRGGBBAA" or "#RGBA") in SVG.

```
\override NoteHead.color = "lightsalmon"
\override Flag.color = "#E30074"
\override Beam.color = "#5e45ad"
\override Rest.color = "#3058"
g'8 \huge r4 a'16 f'
```



- PDF bookmarks are now supported and allow for `\tocItem` entries to appear in the 'table of contents' panel of PDF viewers that support it.
- `\table-of-contents` now accepts a hierarchical structure; `\tocItem` entries may optionally take a symbol (like `\label`) or a dot-separated list of symbols, indicating their position in the score's structure. A side-effect of that feature is that `\tocItem` can no longer take a simple string as its argument; a `\markup` command must be used.

```
\tocItem \markup "Top-level entry"
\tocItem symbol "Named top-level entry"
\tocItem symbol.list "Second-level (child) entry"
```

- Using the new options `-dpng-width` and `-dpng-height` it is now possible to specify the pixel size of PNG output images.
- SVG output is now available through the `--svg` command-line option (or its canonical form `--format=svg`, also shortened as `-fsvg`). Due to its backend's specificity, that option is not yet compatible with other output formats; to get a file in both SVG and PDF, PNG or EPS, a second LilyPond run is required.