LilyPond

Il compositore tipografico per la musica

Cambiamenti

Il team di sviluppo di LilyPond

Questo documento elenca i cambiamenti e le nuove funzionalità della versione 2.22.0 di LilyPond rispetto alla 2.20.

Questo manuale è disponibile in altri formati ed è integrato col resto della documentazione. Maggiori informazioni in Sezione “Manuali” in *Informazioni generali*.

La documentazione completa si trova all’indirizzo `http://lilypond.org/`.
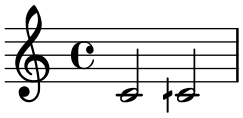
Questo documento è stato posto nel pubblico dominio.

Per la versione di LilyPond 2.22.0

Novità nella notazione musicale
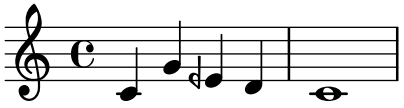
Miglioramenti dell'aspetto delle altezze

- È stato aggiunnto un glifo per le alterazioni corrispondente al codice SMuFL U+E2E3 (alterazioni dell'intonazione Extended Helmholtz-Ellis: aumenta di un quarto di tono non decimale).
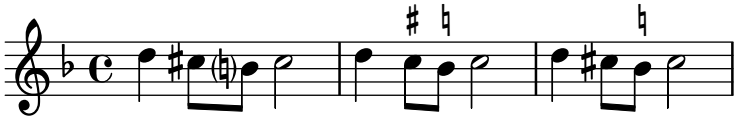


- I nomi delle note per i quarti di tono sono ora disponibili in tutte le lingue di input.

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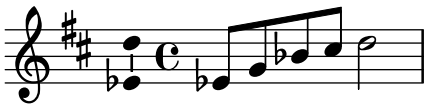


- Impostando `suggestAccidentals` su `'cautionary` solo le alterazioni di cortesia sono trasformate in alterazioni suggerite (`AccidentalSuggestion`). Ciò può servire per distinguere tra il manoscritto originale e le alterazioni editoriali.

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



- È stato aggiunto il comando `\ambitusAfter`, usato per spostare l'ambitus in una posizione diversa.
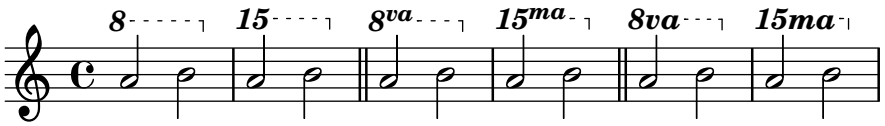
```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



- Le parentesi dell'ottava, per comportamento predefinito, vengono ora introdotte da un numero singolo (per esempio, '8' o '15') e visualizzate in grassetto. Per modificare questo

comportamento è stata introdotta una nuova proprietà `ottavationMarkups`, che ha a disposizione vari elenchi predefiniti di testi formattati:
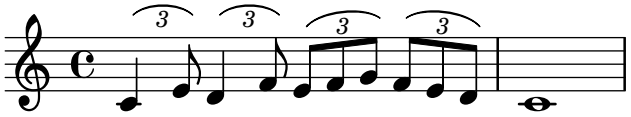
```
\relative c'' {
  \ottava #1
  a'2 b
  \ottava #2
  a'2 b
  \bar "||"
  \set Staff.ottavationMarkups = #ottavation-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
  \bar "||"
  \set Staff.ottavationMarkups = #ottavation-simple-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
}
```
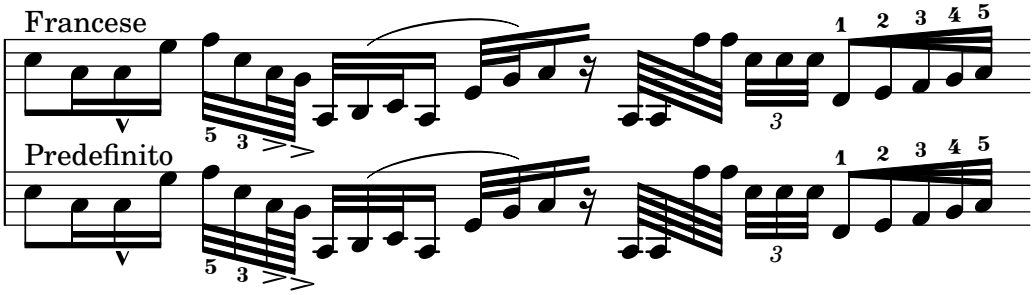


Miglioramenti del ritmo

- I gruppi irregolari ora possono avere le legature di portamento al posto delle parentesi:

```
\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
}
```

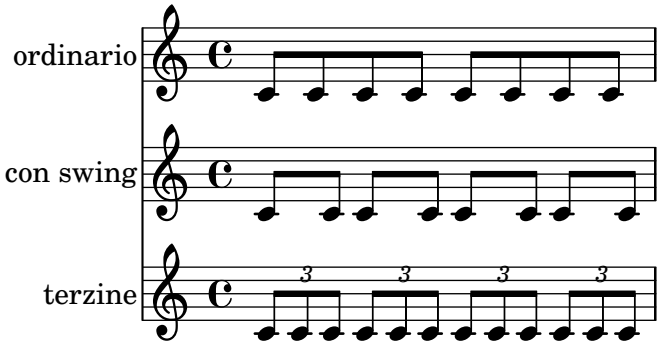


- La disposizione delle travature alla francese (`\override Stem.french-beaming = ##t`) ora si comporta *esattamente* come nella disposizione standard (ovvero quella predefinita) in ogni aspetto (posizionamento della travatura e sistemazione di ciascuna articolazione, diteggiatura, etc.). L'unica differenza rimasta è data dai gambi più interni che non attraversano le travature.

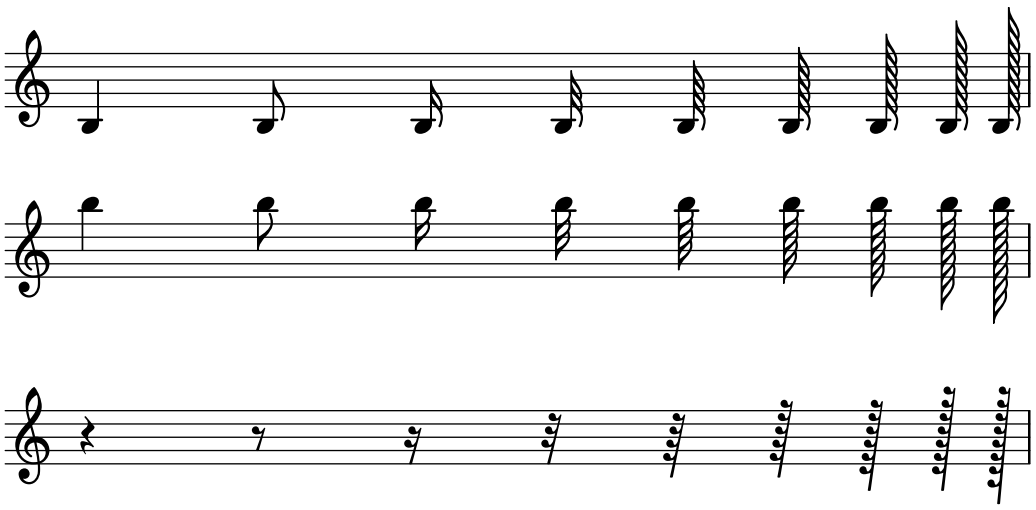


- Lo swing e gli schemi ritmici irregolari ora possono essere applicati alle espressioni musicali fatte di durate normali, potendo così ottenere un'interpretazione ritmica disuguale nel MIDI.

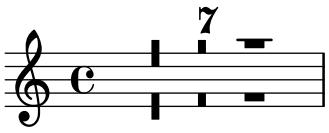
```
\include "swing.ly"
<<
  \new Staff \with {
    instrumentName = "ordinario"
  } \repeat unfold 8 c'8
  \new Staff \with {
    instrumentName = "con swing"
  } \repeat unfold 4 \tripletFeel 8 { c'8 c' }
  \new Staff \with {
    instrumentName = "terzine"
  } \tuplet 3/2 4 \repeat unfold 12 c'8
>>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



- Sono stati aggiunti i glifi dei font per le code uncinate e le pause da 256, 512 e 1024.

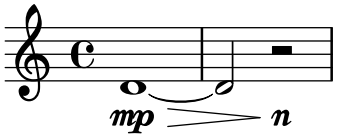


- Il nuovo incisore `Merge_mmrest_numbers_engraver` nasconde i numeri duplicati nelle pause multiple.
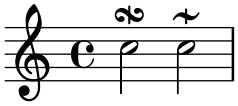


Miglioramenti dei segni espressivi

- È stato aggiunto un comando di dinamica `\n` per *niente*.



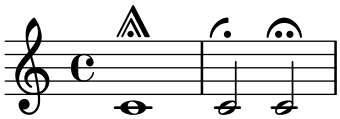
- Sono stati aggiunti due nuovi ornamenti.

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



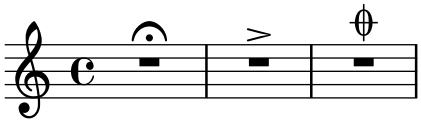
- Sono stati aggiunti un nuovo punto coronato molto breve e le varianti Henze dei punti coronati lunghi e brevi.

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



- I punti coronati e altre articolazioni ora possono essere aggiunti direttamente alle pause multiple. Dunque il comando `\fermataMarkup` è deprecato.

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

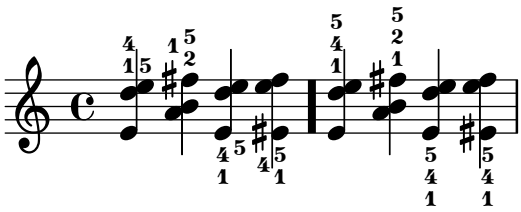


Miglioramenti delle annotazioni editoriali

- L'incisore `New_fingering_engraver` tiene ora in conto della proprietà `X-align-on-main-noteheads` di `self-alignment-interface`. Se impostata su vero (`##t`), tutte le diteggiature orientate in sù (`up`) o in giù (`down`) saranno sistemate su una colonna diritta e allineate alle teste di nota sul lato corretto del gambo:

```
{
  \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>
}
```



Miglioramenti della formattazione del testo

- I font predefiniti sono ora quelli del set URW Core 35, versione 2.0, che sostituiscono i font TeX Gyre. Questi font hanno una legatura per 'Nr.'; vedi Sezione "Tipi di carattere in dettaglio" in *Guida alla Notazione* per soluzioni in grado di evitarla localmente e globalmente.
- Il comando di markup `\note` ora prende come primo argomento una durata invece di una stringa:

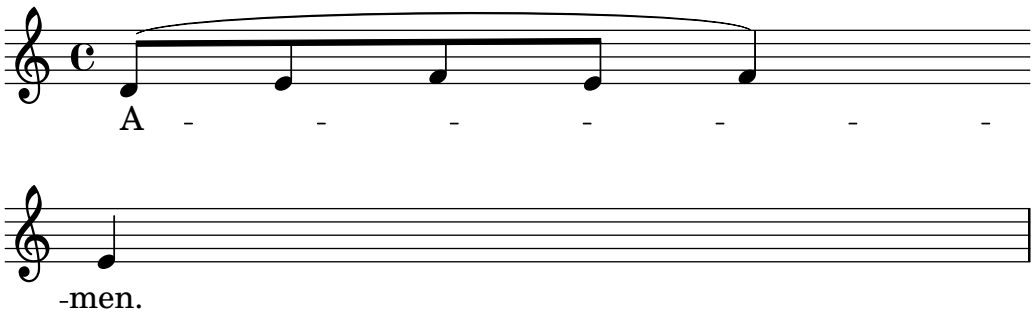
```
\markup {
  \override #'(style . cross) {
    \note {4..} #UP
  }
  \hspace #2
  \note {\breve} #0
}
```



Novità per la notazione specialistica

Miglioramenti della musica vocale
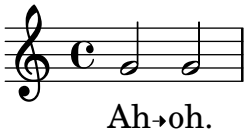
- Un trattino del testo vocale ora può essere ripetuto all'inizio di un sistema che inizia con una nuova sillaba:



- Un cambiamento graduale di vocale (o consonante sostenuta) può essere indicato aggiungendo una transizione di vocale tra le sillabe del testo col comando `\vowelTransition`.

```
{ g'2 g' }
```

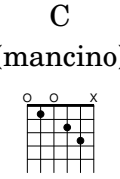
```
\addlyrics { Ah \vowelTransition oh. }
```



Miglioramenti degli strumenti a corda con e senza tasti

- I diagrammi dei tasti ora possono essere resi per i mancini, impostando `handedness` su `LEFT` (come sottoproprietà di `fret-diagram-details`).

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

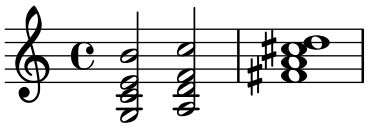


- Alcuni diagrammi dei tasti dell'ukulele sono stati corretti e sono stati aggiunti anche nuovi diagrammi.

Migloramenti della notazione per accordo

- È ora possibile creare automaticamente il rivolto di un accordo oppure cambiarne il *voicing* con note abbassate.

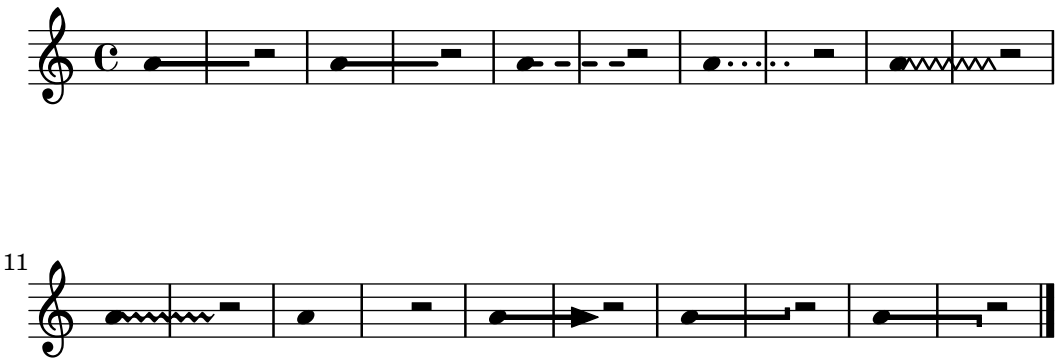
```
\chordmode {
  \dropNote 2 {
    c2:maj7 d:m7
  }
  \invertChords 1 d1:maj7
}
```



- Le vecchie funzioni per i nomi degli accordi `banter-chord-names` e `jazz-chord-names` sono state tolte dal codice sorgente principale, così come le proprietà `chordNamesExceptionsFull` e `chordNamesExceptionsPartial`. Come parte di quella riscrittura, i power chord sono ora inclusi nelle eccezioni predefinite; appariranno correttamente (con un apice come in tutti gli altri accordi) senza bisogno di ulteriori modifiche, rendendo quindi obsoleto il comando predefinito `\powerChords`. Chi fosse interessato alle vecchie funzioni può trovare una loro copia (in qualche modo funzionante) nel frammento `chord-names-alternative.ly`.

Miglioramenti per la musica contemporanea

- È disponibile un nuovo grob: `DurationLine` (linea di durata). Continua un grob ritmico con una linea, terminandola al grob ritmico successivo. Gli stili possibili sono `'beam`, `'line`, `'dashed-line`, `'dotted-line`, `'zigzag`, `'trill` e `'none`. La linea di durata può terminare con un gancio (solo beam-style) o con una freccia.



Novità relative a input e output

Miglioramenti dei file di input

- Come annunciato per la versione 2.17.3 quasi otto anni fa, l'opzione `'relative-includes` è ora abilitata per impostazione predefinita; i file inclusi che contengono un comando `\include` per conto loro devono considerare il percorso relativo alla propria directory corrente invece che a quella del file principale. Questo comportamento può comunque essere disattivato impostando `'relative-includes` su `#f`, come opzione da linea di comando o usando `ly:set-option` nei file sorgenti.

- `\compressFullBarRests` è stato rinominato `\compressEmptyMeasures`, per evitare la possibile confusione con `\compressMMRests`. Analogalmente, `\expandFullBarRests` si chiama ora `\expandEmptyMeasures`.

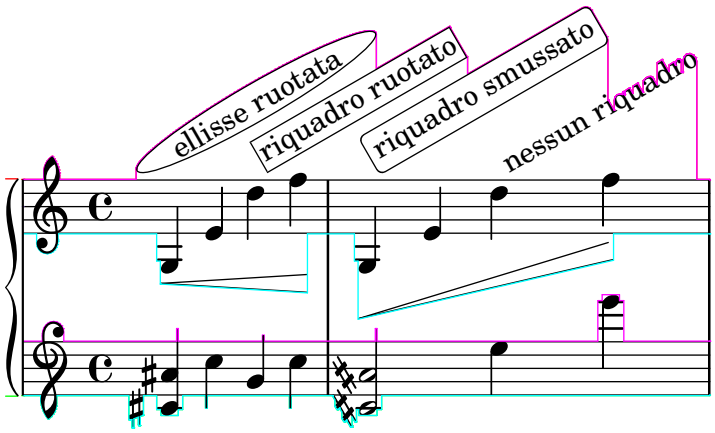
- Il comando `\partcombine`, così come tutti i comandi, sottoprocedure e nomi di proprietà che iniziano con `partCombine`, ora si scrivono con una C maiuscola, per esempio `\partCombine`, `\partCombineApart` etc. Il comando `\autochange` viene anch'esso scritto con la maiuscola come `\autoChange`.

- Tutte le lingue di input (dichiarazione `\language`) si possono inserire usando le loro corrette lettere UTF-8 (inclusi quindi i caratteri speciali). I nomi mancanti `català` e `português` sono ora disponibili in aggiunta ai nomi originali `catalan` e `portugues`.

- LilyPond per Windows (MinGW) può gestire i nomi file in Unicode su Windows 10 1903 e successivi.
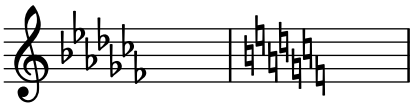
Miglioramenti dell'output

- I profili (*skylines*) tengono ora in considerazione la proprietà `rotation` degli oggetti di formattazione. Per esempio, ruotando una forcella di crescendo con `\override Hairpin.rotation = #'(15 0 0)`, ora avrà realmente effetto sui profili e dunque aiuterà a migliorare la spaziatura.



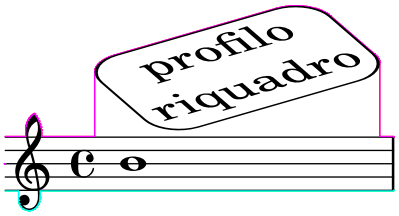
- Aggiunto un leggero padding tra i glifi di bequadro che quasi si toccano agli angoli degli annullamenti delle alterazioni poste nell'armatura di chiave.

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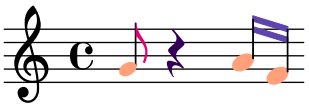


- I profili dei riquadri ora riflettono il reale contorno del riquadro, inclusi gli angoli arrotondati e la rotazione.

```
#(ly:set-option 'debug-skylines #t)
{
  b'1^\markup \scale #'(2 . 1)
              \rotate #30
              \override #'(corner-radius . 3)
              \rounded-box \center-column { profilo riquadro }
}
```



- I colori in stile CSS ora possono essere usati direttamente come stringhe di testo, sia coi nomi di colore predefiniti (come con la funzione `css-color` già disponibile), sia coi codici colore esadecimale preceduti da `#`. Tutti gli stampi (stencil) che supportano una proprietà `color` ora accettano sia una lista che una stringa; nel secondo caso, quella stringa viene usata direttamente nell'output SVG. Ciò permette di usare la trasparenza sul canala alfa (inserita come "#RRGGBBAA" o "#RGBA") nel formato SVG.

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



- Sono ora supportati i segnalibri PDF. Gli elementi `\tocItem` appaiono nel pannello 'indice' dei lettori PDF che lo supportano.
- `\table-of-contents` ora accetta una struttura gerarchica; gli elementi `\tocItem` possono prendere (facoltativamente) un simbolo (come `\label`) o un elenco di simboli separati dal punto, che indichino la loro posizione nella struttura della partitura. Un effetto collaterale di questa funzionalità è che `\tocItem` non può più prendere una semplice stringa come suo argomento; bsogna invece usare un comando `\markup`.

```
\tocItem \markup "Elemento di livello superiore"
\tocItem symbol "Elemento di livello superiore con nome"
\tocItem symbol.list "Elemento di secondo livello (figlio)"
```

- È ora possibile specificare la dimensione in pixel delle immagini di output PNG con le nuove opzioni `-dpng-width` e `-dpng-height`.
- L'output SVG si può ora ottenere attraverso l'opzione da linea di comando `--svg` (o la sua forma canonica `--format=svg`, abbreviata anche come `-fsvg`). A causa della specificità del suo backend, questa opzione non è ancora compatibile con altri formati di output; per produrre file sia SVG che PDF, PNG o EPS, è richiesta una seconda compilazione da parte di LilyPond.