LilyPond

Le système de gravure musicale

Nouveautés

L'équipe de développement de LilyPond

Ce document recense les modifications et les nouvelles fonctionnalités de LilyPond pour la version 2.27.5 (depuis la 2.26).

Pour connaître la place qu'occupe ce manuel dans la documentation, consultez la page Section "Manuels" dans *Informations générales*.

Si vous ne disposez pas de certains manuels, la documentation complète se trouve sur `https://lilypond.org/`.

Ce document a été placé dans le domaine public ; en France, les auteurs renoncent à tous leurs droits patrimoniaux.

Pour LilyPond version 2.27.5

Note : Chaque nouvelle version de LilyPond peut comporter des changements de syntaxe, ce qui peut exiger de modifier les fichiers que vous avez écrits avec des versions précédentes, de telle sorte qu'ils restent fonctionnels avec les nouvelles versions. Afin de mettre à jour vos fichiers, il est **fortement conseillé** d'utiliser l'utilitaire `convert-ly` distribué avec LilyPond et qui est abordé dans Voir Section "Mise à jour avec convert-ly" dans *Utilisation des programmes*. `convert-ly` est capable de réaliser la plupart des modifications de syntaxe automatiquement. Les utilisateurs de Frescobaldi peuvent lancer `convert-ly` directement à partir du menu de Frescobaldi en faisant « Outils > Mettre à jour avec convert-ly… ». D'autres environnements prenant en charge LilyPond sont susceptibles de fournir un moyen graphique de lancer `convert-ly`.

Table des matières

Modifications majeures de LilyPond 1

Nouveautés en matière de notation musicale 2

Améliorations de la représentation des hauteurs 2
Améliorations en matière de rythme 2
Améliorations en matière d'expressivité 3
Améliorations en matière de reprises 3
Améliorations en matière d'annotations éditoriales 3
Améliorations en matière de fontes et de mise en forme du texte 3

Nouveautés en matière de notation spécialisée 5

Améliorations pour la musique vocale 5
Améliorations pour instruments à portées multiples 6
Améliorations pour les cordes non frettées 6
Améliorations pour les cordes frettées 6
Améliorations pour la notation des percussions 6
Améliorations pour la notation des instruments à vent 6
Améliorations pour la notation des accords 6
Améliorations pour les notations anciennes 6
Améliorations pour les musiques du monde 7

Autres améliorations diverses 8

Modifications majeures de LilyPond

- Les expressions `book` et `bookpart` sont désormais indépendantes l'une de l'autre et sont typées différemment. Ceci peut se tester avec `ly:book?` et `ly:bookpart?`. Un prédicat permettant de tester duquel il s'agit – comme le faisait auparavant `ly:book?` – est disponible sous le nom de `ly:book-or-bookpart?`.

  Il est désormais possible d'utiliser directement, plutôt que des blocs `book` ou `bookpart`, des identifiants de *book* et *bookpart* ainsi que des appels à des fonctions syntaxiques en Scheme définies par `define-scheme-function`.
- Les blocs `\book` et `\bookpart` contiennent désormais leur propre portée de variables. Cette accessibilité est active dans l'entièreté du bloc et recense toutes les variables définies au sein de ce bloc. Par voie de conséquence, les définitions de variables sont maintenant permises au sein des blocs `\book` et `\bookpart`, ce qui facilite la restructuration des fichiers LilyPond indépendants en les intégrant dans des ouvrages ou chapitres. Ces variables sont inaccessibles en dehors du bloc qui les contient. L'interface publique avec le module contenant les variables d'un `\book` ou d'un `\bookpart` est accessible de manière programmatique à l'aide de la nouvelle fonction Scheme `ly:book-scope`; les fonction d'accès `ly:book-lookup` et `ly:book-set-variable!` sont aussi fournies.
- `\book` et `\bookpart` peuvent désormais contenir, en plus d'un bloc `\paper`, de blocs de définition de sortie `\layout` et `\midi`. Ils récupèrent leurs réglages initiaux à partir de la portée active en cours. Lorsqu'un *bookpart* dispose de son propre bloc `\layout`, il fournit les réglages de gravure par défaut de toutes les partitions qu'il inclut et qui n'ont pas leur propre définition de sortie. Il en va de même pour un *book*. Dans la mesure où uns *score* ne produit du MIDI que lorsqu'il inclut explicitement un bloc `\midi`, il n'y a pas de comportement par défaut identique lors du traitement.

Nouveautés en matière de notation musicale

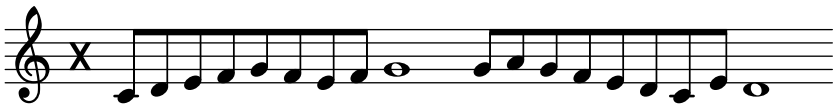
Améliorations de la représentation des hauteurs

Rien pour l'instant.

Améliorations en matière de rythme
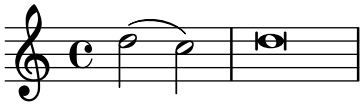
- La nouvelle commande `\senzaMisura` institue un « temps libre » et sur option imprime un métrique en « X ». Ce signe est accessible par `\senzaMisuraTimeSignatureX` et invalidé par `\senzaMisuraTimeSignatureOff`.

```
\fixed c' {
  \senzaMisuraTimeSignatureX
  \senzaMisura
  c8 d e f g f e f g1
  g8 a g f e d c e d1
}
```



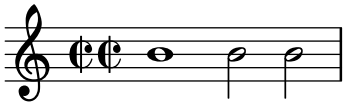
- La nouvelle commande `\measure` modifie la longueur de la mesure afin qu'elle corresponde à la durée d'une musique donnée puis revient à la normale.

```
\fixed c'' {
  d2( c)
  \measure { d\breve }
}
```
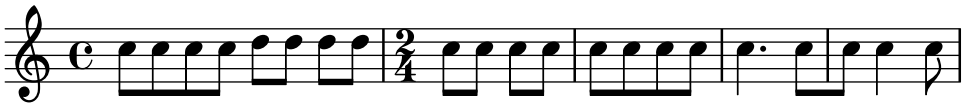


- La nouvelle commande `\measureRemainder` est similaire à `\measure`, à ceci près qu'elle peut s'utiliser en cours de mesure.
- La nouvelle commande `\setMeasureLengthFromHere` simplifie la création de mesures irrégulières en réglant la longueur de la mesure relativement à la position dans la mesure actuelle. Ceci vient en complément de `\partial`, qui modifie la position dans la mesure par rapport à la longueur de cette mesure. La nouvelle commande `\setDefaultMeasureLength` réinitialise la longueur de mesure à la valeur déterminée par la métrique.
- La nouvelle instruction `\premeasure` *`musique`* constitue une abréviation de `\partial` *`durée musique`* qui calcule automatiquement la durée.
- Par défaut, les mesures alternées composées de 4/4 et de 2/2 s'affichent dans un style ancien avec plusieurs symboles « C » ou « C barré ».

```
{
  \time #'((2 . 2) (2 . 2))
  b'1 2 2
}
```



- L'algorithme de ligature automatique peut se voir obligé d'interrompre en un point particulier une ligature créée automatiquement grâce à un `\beamBreak`. De manière similaire, deux notes adjacentes peuvent être forcées de se ligaturer à l'aide d'un `\noBeamBreak` :

```
\relative {
  c''8 c c c d d \beamBreak d d
  \time 2/4
  c8 8 8 8
  8 8 \noBeamBreak 8 8
  4. 8 \noBeamBreak
  8 4 8
}
```



Améliorations en matière d'expressivité

Rien pour l'instant.

Améliorations en matière de reprises
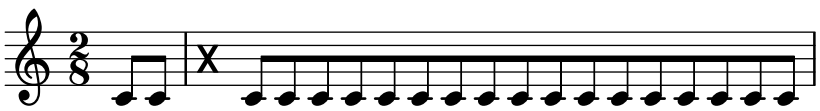
Rien pour l'instant.

Améliorations en matière d'annotations éditoriales

- La fonction fonction `parenthesize-stencil` accepte un argument optionnel afin d'agrandir les parenthèses.
- La fonction fonction `bracketify-stencil` accepte un argument optionnel afin d'agrandir les crochets.

Améliorations en matière de fontes et de mise en forme du texte

- Un signe égal a été ajouté à la fonte Emmentaler.
- Une métrique en « X » a été ajoutée à la fonte Emmentaler. Elle peut optionnellement s'imprimer lorsque le mètre est annulé par la commande `\senzaMisura`. L'utilisation de ce signe répond à la commande `\senzaMisuraTimeSignatureX` ; il est désactivé par `\senzaMisuraTimeSignatureOff`.



- Des variantes des glyphes textuels pour les symboles de métriques en 4/4 (*common time*) et 2/2 (*cut time*) ont été ajoutés à la fonte Emmentaler. Il sont accessibles soit directement par leurs caractères Unicode, soit à l'aide des commandes de *markup* `\text-common-time` et `\text-cut-time`.

Common time (𝐂), cut time (𝐂).

- Les fontes peubvent se changer au sein même d'un *markup* à l'aide de `\font-select`. Les noms de fonte pour les familles à empattement (*serif*), sans empattement (*sans*) et monospace (*typewriter*) sont indépendants les uns des autres.

```
\markup \font-select #'((sans . "DejaVu Sans")
                        (typewriter . "DejaVu Sans Mono"))
{
  \sans \line { Sans-serif font. }
  \typewriter \line { Typewriter font. }
  \line { Default serif font. }
  \font-select #'(serif . "DejaVu Serif")
  \line { Changed serif font. }
}
```
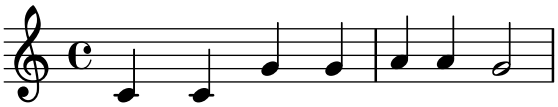
Sans-serif font. Typewriter font. Default serif font. Changed serif font.

Nouveautés en matière de notation spécialisée

Améliorations pour la musique vocale

- Les numéros de couplet dans les paroles sont désormais créés par la commande dédiée `\stanza ...` qui remplace syntaxe `\set stanza = ...` jusqu'ici requise.

```
\relative {
  c'4 c g' g
  a a g2
}
\addlyrics {
  \stanza "1."
  Twin -- kle, twin -- kle, lit -- tle star
}
\addlyrics {
  \stanza "2."
  When the bla -- zing sun is gone
}
```
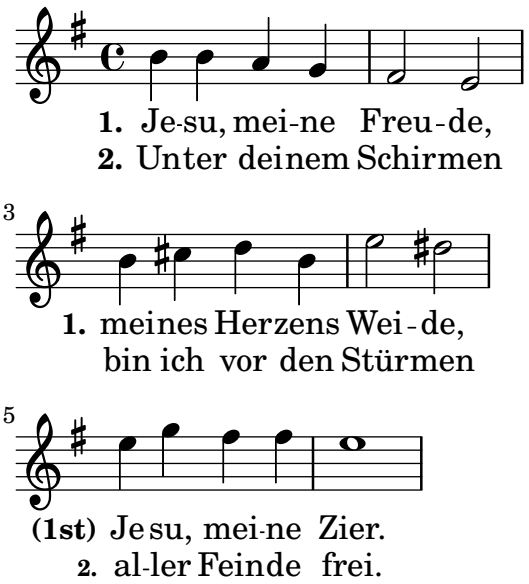


1. Twin-kle, twin-kle, lit-tle star
2. When the bla-zing sun is gone

- En matière de paroles, le numéro de couplet peut se répéter automatiquement en tête de chaque ligne. Cette fonctionnalité s'obtient après l'activation de la propriété `stanzaReminders` :

```
\new Voice \relative {
  \key e \minor
  b'4 b a g
  fis2 e \break
  b'4 cis d b
  e2 dis \break
  e4 g fis fis
  e1
}
\addlyrics {
  \stanza "1."
  \set stanzaReminders = ##t
  Je -- su, mei -- ne Freu -- de,
  mei -- nes Her -- zens Wei -- de,
  \set stanzaReminderText = "(1st)"
  Je -- su, mei -- ne Zier.
}
\addlyrics {
  \stanza "2."
  Un -- ter dei -- nem Schir -- men
  bin ich vor den Stür -- men
  \set stanzaReminders = ##t
  \set stanzaReminderText = #make-tiny-markup
  al -- ler Fein -- de frei.
```

```
}
```



Améliorations pour instruments à portées multiples

Rien pour l'instant.

Améliorations pour les cordes non frettées

Rien pour l'instant.

Améliorations pour les cordes frettées

Rien pour l'instant.

Améliorations pour la notation des percussions

Rien pour l'instant.

Améliorations pour la notation des instruments à vent

Rien pour l'instant.

Améliorations pour la notation des accords

- Le nom d'accord répondant à la saisie `x:1/x` est désormais rendu en « N.C./*x* ».

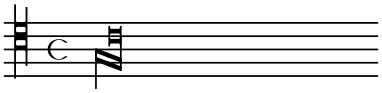
```
\chords { d:1/d }
```



Améliorations pour les notations anciennes

- Les ligatures mensurales blanches prennent désormais en charge une longue finale ascendante renversée, tiennent compte des ajustements dans plus de cas, et gèrent les erreurs plus facilement.

```
\score {
  \new PetrucciStaff {
    \clef "petrucci-c4"
    \[ g\breve f \tweak ligature-pes ##t c'\longa \]
  }
}
```



- La commande \~ est une alternative au instructions \pes et \flexa qui, en interne, font la même chose. Bien qu'introduite en 2003, cette commande était jusqu'à présent absente de la documentation et a connu plusieurs dysfonctionnements.

```
\new VaticanaScore {
  \new VaticanaVoice \relative {
    \[ a \pes b \flexa g \]
  }
}
```



```
\new VaticanaScore {
  \new VaticanaVoice \relative {
    \[ a \~ b \~ g \]
  }
}
```
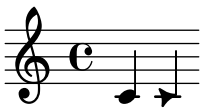


Améliorations pour les musiques du monde

Rien pour l'instant

Autres améliorations diverses

- La nouvelle commande `\verboseBarNumbers` simplifie les exemple éducatifs pour lesquels les numéros de mesure sont importants. Elle s'avère utile aussi en matière de débogage.
- Les nouvelles commandes `\startGradualTempoChange` et `\stopGradualTempoChange` marquent le début et la fin d'un changement progressif de tempo. Elles ne produisent leurs effets, pour l'instant, que sur des sorties MIDI.
- Les fonctions suivantes, qui avaient été dépréciées avec la version 2.13.19, ont été supprimées : `\deprecatedcresc`, `\deprecatedendcresc`, `\deprecateddim` et `\deprecatedenddim`.
- La propriété `glyph-name` d'un objet `NoteHead` contient désormais un nom pleinement qualifié de glyphe, comme le dit sa documentation. Il n'est donc plus nécessaire de modifier un stencil pour juste obtenir un glyphe de tête de note différent ; régler cette propriété est suffisant.

```
{
  c'4
  \once\override NoteHead.glyph-name = "noteheads.s2triangle"
  c'4
}
```



Comme effet secondaire, la fonction Scheme `note-head::calc-glyph-name` n'est plus nécessaire et sera supprimée après une période de dépréciation. Selon le besoin, le recours à cette fonction pourra être remplacé par un accès à la propriété `glyph-name` d'un objet `NoteHead` particulier, ou en faisant appel à `select-head-glyph` pour un style spécifique de tête de note avec sa durée.