

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 2.23.5 (depuis la 2.22).

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 2.23.5

Nouveautés en matière de notation musicale

Améliorations de la représentation des hauteurs

- La prise en charge d'altérations alternatives s'améliore. Grâce à la propriété `alterationGlyphs` des contextes de niveau portée, peuvent se définir globalement les glyphes à utiliser pour tous les objets – voir Section “Glyphes d'altération alternatifs” dans *Manuel de notation*.



- Dans le cadre de la fonte Emmentaler, les têtes de notes dont l'aspect est identique et pour lesquelles la seule différence réside dans l'orientation de leur hampe ont été consolidées en un glyphe unique. Par exemple, les glyphes `noteheads.u2triangle` et `noteheads.d2triangle` ont été remplacés par le seul glyphe `noteheads.s2triangle`. Les paires de tête de note ayant un aspect différent selon l'orientation de la hampe conservent leur distinction.

Par ailleurs, la propriété `stem-attachment` des objets graphiques `NoteHead` renvoie le point d'attachement réel selon l'orientation de hampe en lieu et place d'un hypothétique point d'attachement d'une hampe ascendante.

- Les crochets d'octaviation peuvent s'appliquer à une seule voix plutôt qu'à l'intégralité de la portée. Ceci demandait auparavant quelques circonvolutions.

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



Améliorations en matière de rythme

- Les numéros de mesure peuvent se présenter au centre de la mesure comme il est d'usage dans les musiques de film.

```
\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' } }
>>

```

- Les compteurs de mesures prennent en considération les mesures à compter et les alternatives.

- `\numericTimeSignature` et `\defaultTimeSignature` s'appliquent désormais en même temps à toutes les portées – plus précisément à toutes les portées d'un même contexte `Timing` – afin de correspondre au comportement de `\time`.
- Les événements attachés à des notes, tels que nuances ou articulations, peuvent être différés d'une durée arbitraire à l'aide de `\after`. Ceci permet de simplifier de nombreuses situations qui jusqu'alors demandaient de recourir à une polyphonie et des silences invisibles.

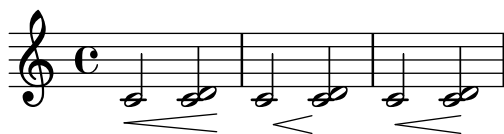
```

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

```

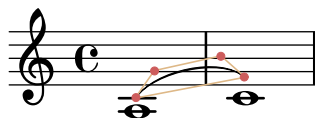
Améliorations en matière d'expressivité

- Les terminaisons des soufflets peuvent désormais s'aligner sur la gauche, au centre ou sur la droite des `grobs NoteColumn` à l'aide d'une dérogation à la propriété `endpoint-alignments`.



- La nouvelle commande `\vshape` se comporte comme `\shape`, à ceci près qu'elle affiche les points de contrôle et le polygone qui les enveloppe, afin de faciliter les ajustements.

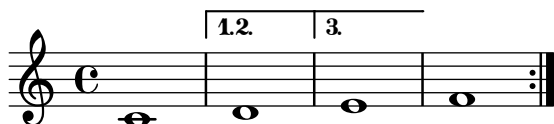
```
{ a1\vshape #'((0 . 0) (0 . 0.5) (0 . 0.9) (0 . 0.4))^( c'1) }
```



Améliorations en matière de reprises

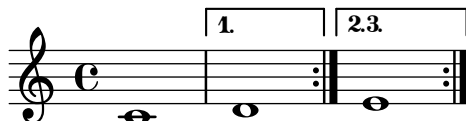
- Les alternatives peuvent se placer ailleurs qu'après la fin du fragment répété.

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



- Les numéros d'alternative peuvent se définir à l'aide de la commande `\volta`.

```
\repeat volta 3 c'1 \alternative { \volta 1 d' \volta 2,3 e' }
```



- La commande `\volta` supprime la musique lorsque la répétition est expansée.
- La commande `\unfolded` ajoute la musique lorsque la répétition est expansée.

Améliorations en matière de notation sur la portée

- L'ajout du `Melody_engraver` à un contexte `Voice` prend désormais en compte le sens de la mélodie pour orienter la hampe de la note sur la ligne médiane. Il fallait auparavant apporter une dérogation spécifique à la propriété `Stem.neutral-direction`.

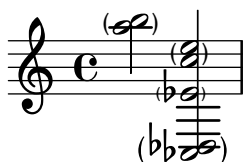
```
\new Voice \with {
  \consists Melody_engraver
}
\relative c'' {
  \autoBeamOff
  g8 b a e g b a g |
  c b d c b e d c |
}
```



La propriété de contexte `suspendMelodyDecisions` permet de désactiver temporairement ce comportement, tout comme le faisait `\override Stem.neutral-direction = #DOWN`.

Améliorations en matière d'annotations éditoriales

- Il est désormais possible de placer un accord entre parenthèses. Toutefois, la taille de la fonte utilisée pour les parenthèses devra être ajustée manuellement.



- Les objets étendus peuvent être mis entre parenthèses.



- Une version « temporelle » de la commande `\parenthesize` est disponible. Elle prend en considération le chemin vers un objet graphique : `\parenthesize NomGrob` ou `\parenthesize ContextName.GrobName`. Cette commande se comporte comme un `\once \override`. Cette interface vient en complément de la forme déjà disponible `\parenthesize événement`, de manière identique à `\footnote`.

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



- En matière de basse chiffrée, il est désormais possible de placer les altérations entre crochets.

```
7 [b]5 [b]5
[#13]
```

Améliorations en matière de mise en forme du texte

- La nouvelle commande de `markup string-lines` permet de couper une chaîne à un caractère donné. La coupure intervient par défaut au saut de ligne, et les éventuels espaces sont ignorés. La liste résultante de `markups` peut faire l'objet d'une mise en forme particulière. Cette fonctionnalité constitue un moyen tout à fait adapté pour ajouter des couplets à une chanson.

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!

Nouveautés en matière de notation spécialisée

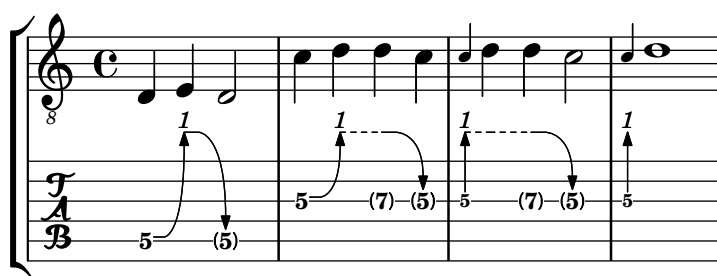
Améliorations pour les cordes frettées ou non

- Ajout des accordages pour banjo `banjo-double-c` et `banjo-double-d`.
- Le nouvel objet graphique `FingerGlideSpanner` permet d'indiquer le glissé d'un doigt sur une corde passant d'une position à une autre. Il peut se présenter sous différentes formes selon le style adopté ; l'image ci-dessous affiche les styles `line`, `stub-left`, `stub-right` et `stub-both`.



D'autres styles sont disponibles : `dashed-line`, `dotted-line`, `zigzag`, `trill`, `bow` et `none`.

- Le nouvel objet graphique `BendSpanner` permet, dans le cadre d'un `TabStaff`, d'indiquer une désinace. En plus de l'apparence par défaut sont disponibles les styles `'hold`, `'pre-bend` et `'pre-bend-hold`.



Nouveautés en matière d'entrée et sortie

Améliorations pour la structure

- Une même définition d'objet graphique peut désormais s'utiliser pour créer des *grobs* de classe différente (`Item`, `Spanner`, `Paper_column`, `System`). Dans le cadre de cette évolution, les types de *grob* `FootnoteItem` et `FootnoteSpanner` ont été rassemblés en un unique type `Footnote`. De la même manière, `BalloonTextSpanner` et `BalloonTextItem` ont été unifiés en `BalloonText`.

Lorsque la définition du *grob* ne requiert pas de classe, les graveurs doivent décider de la classe à utiliser pour créer le *grob*. Pour ceux qui créent leurs propres graveurs en `Scheme`, cela signifie qu'il faudra utiliser soit `ly:engraver-make-item`, soit `ly:engraver-make-spanner`. La fonction utilitaire `ly:engraver-make-sticky` permet de prendre en charge le cas des *sticky grobs* tels que notes de bas de page et infobulles. Elle crée un *grob* de la même classe qu'un autre et gère la parenté et les attachements.

Améliorations pour les titrages et entêtes

- La syntaxe permettant de gérer des conditions pour les *markups* a gagné en flexibilité et en simplicité. Elle utilise les nouvelles commandes de *markup* `\if` et `\unless`. Voici quelques uns des changements apportés :

Syntaxe version 2.22

```
\on-the-fly #first-page ...
\non-the-fly
#not-part-first-page
...
```

Syntaxe version 2.24

```
\if \on-first-page ...
\nunless \on-first-page-of-part ...
```

`\on-the-fly #(on-page n) ...` `\if \on-page #n ...`


Améliorations pour les fichiers résultants

- La transparence des couleurs est prise en charge lors de la génération de PostScript et de PDF. L'utilisation de cette fonctionnalité requiert une version de Ghostscript égale ou supérieure à la 9.53.
- `set-default-paper-size` et `set-paper-size` acceptent désormais des tailles de papier personnalisées.

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

- `lilypond-book` prend en charge deux nouvelles options pour le traitement des extraits : `paper-width` et `paper-height` permettent de déterminer une taille particulière de papier.

Nouveautés en matière d'interfaces et fonctions internes

- Le script `lilypond-book` autorise désormais l'utilisation d'accolades dans les arguments des commandes `\lilypond` (pour L^AT_EX) et `@lilypond` (pour Texinfo).
- `lilypond-book` ajoute désormais le répertoire courant comme dernière entrée pour la recherche des fichiers inclus, au lieu de le placer en premier dans la liste des chemins spécifiés. Ainsi, les fichiers situés dans les dossiers d'inclusion prennent le pas sur ceux du même nom dans le répertoire courant. L'effet ne sera visible que dans le cas où existent des fichiers du même nom de part et d'autre.
- `lilypond-book` prend en charge la nouvelle option `inline` pour le traitement d'extraits musicaux. Ceci permet de faire apparaître des bribes de partition, comme  au sein même d'un paragraphe de texte.
- Les traducteurs définis en Scheme et utilisables à la fois dans le cadre d'un '`\layout`' et d'un '`\midi`' peuvent désormais être créés à l'aide de `make-translator`. Les exécutants définis en Scheme, qui ne peuvent s'utiliser de dans un '`\midi`' se créent à l'aide de `make-performer`. Ces macros fonctionnent tout comme la macro préexistante `make-engraver` qui permet de créer un graveur, uniquement utilisable dans un '`\layout`'.