

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 2.25.4 (depuis la 2.24).

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 2.25.4

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`.

Modifications majeures de LilyPond

- Les marges sont désormais plus larges, suivant ainsi les mises en pages de nombreux éditeurs, et conformément aux recommandations d'Elaine Gould.

Pour retrouver les mêmes réglages que précédemment, notamment dans le but de conserver la mise en page après mise à jour d'une partition à la version 2.25.4, il suffit d'ajouter le code suivant :

```
\paper {
  top-margin = 5\mm
  bottom-margin = 10\mm
  top-system-spacing.basic-distance = 1
  top-markup-spacing.basic-distance = 0
  left-margin = 10\mm
  right-margin = 10\mm
  inner-margin = 10\mm
  outer-margin = 20\mm
  binding-offset = 0\mm
}
```

- Au lieu de générer des sorties PostScript ou SVG par lui-même, LilyPond peut désormais utiliser la bibliothèque Cairo pour produire ses résultats. Il est ici fait référence au « moteur Cairo » qui peut être activé par l'option `-dbackend-cairo` en ligne de commande. Cette fonctionnalité est opérationnelle pour tous les formats de sortie (PDF, SVG, PNG, PostScript) et apporte vitesse et amélioration du rendu SVG en particulier. Néanmoins, les fonctionnalités des moteurs par défaut ne sont pas encore toutes implémentées. Sont entre autres absent le plan des PDF, l'option `-dembed-source-code` pour le PDF et la propriété `output-attributes` pour le SVG.

Nouveautés en matière de notation musicale

Améliorations de la représentation des hauteurs

Rien pour l'instant.

Améliorations en matière de rythme

- La nouvelle option `span-all-note-heads` permet aux crochets de n-olets d'embrasser toutes les têtes de notes (pas seulement les hampes) comme recommandé par Gould et Ross.



- Grâce à la propriété `printInitialRepeatBar`, il est désormais possible d'afficher automatiquement une barre de reprise même lorsqu'elle intervient en début de pièce.



Améliorations en matière d'expressivité

- Sont désormais disponibles deux variantes du signe de respiration : `'altcomma'` et `'raltcomma'`. Ces glyphes représentent respectivement les anciens galbes de « lcomma » et « rcomma » avant leur changement pour un galbe plus courant.

```
{
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.raltcomma" }
  f'2 \breathe f' |
}
```



Améliorations en matière de reprises

Rien pour l'instant.

Améliorations en matière d'annotations éditoriales

- Les objets graphiques `NoteName` sont désormais centrés horizontalement par défaut.

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

- La syntaxe permettant de modifier les fontes musicales et textuelles a changé. Au lieu de

```
\paper {
  #(define fonts
    (set-global-fonts
      #:music "Nom de la fonte musicale"
      #:brace "Nom de la fonte musicale d'accolades"
```

```

        #:roman "Nom de la fonte romaine"
        #:sans "Nom de la fonte sans empattements"
        #:typewriter "Nom de la fonte monospace"))
    }
ou
\paper {
  #(define fonts
    (make-pango-font-tree
      "Nom de la fonte romaine"
      "Nom de la fonte sans empattements"
      "Nom de la fonte monospace"
      factor))
  }

```

la syntaxe consacrée est dorénavant

```

\paper {
  fonts.music = "Nom de la fonte musicale"
  fonts.roman = "Nom de la fonte romaine"
  fonts.sans = "Nom de la fonte sans empattement"
  fonts.typewriter = "Nom de la fonte monospace"
}

```

Contrairement aux anciennes pratiques, la nouvelle syntaxe n'interfère en rien dans la taille des fontes, qui doit se gérer séparément à l'aide de `set-global-staff-size` ou `layout-set-staff-size`.

La liste associative ne comporte pas de clé `brace` ; les glyphes d'accolade sont désormais toujours pris dans la fonte musicale. Il est néanmoins possible d'y déroger en utilisant une famille de fontes supplémentaire, comme dans l'exemple suivant (la fonte LilyJAZZ doit alors être disponible) :

```

\paper {
  fonts.music-alt = "lilyjazz"
}

\layout {
  \context {
    \Score
    \override SystemStartBrace.font-family = #'music-alt
  }
}

\new PianoStaff <<
  \new Staff { c' }
  \new Staff { c' }
>>

\markup \override #'(font-family . music-alt) \left-brace #20

```

- La commande de *markup* `\lookup` n'est désormais disponible que pour les accolades ; pour les autres glyphes, c'est la commande `\musicglyph` qu'il faut utiliser. Au lieu de `\lookup`, il vaut d'ailleurs mieux lui préférer `\left-brace`.
- Lorsqu'une fonte musicale est utilisée dans un *markup* – typiquement pour une indication de nuance – et qu'un glyphe en était absent, celui-ci était rendu dans une fonte textuelle normale. Ceci n'est plus le cas, et un avertissement est alors émis quant au glyphe manquant.

Afin d'utiliser une fonte textuelle, il faut utiliser la commande de *markup* `\text`, comme ici par exemple.

```
dolceP =
#(make-dynamic-script
#{
  \markup {
    \text \normal-weight dolce
    p
  }
#})

{ c'\dolceP }
```



- Les petites capitales s'obtiennent désormais en réglant `font-variant` sur `small-caps`, plutôt qu'en fixant `font-shape` à `caps`. Dans la mesure où la raison d'être de `font-shape` est de pouvoir accéder à l'italique, ce changement rend possible l'utilisation conjointe de petites capitales et de l'italique.

Small caps in italics

Nouveautés en matière de notation spécialisée

Autres améliorations diverses

- L'inclusion d'images PNG est désormais possible à l'aide de la commande de *markup* `\image`. Ceci vient en supplement de la commande `\epsfile` pour les images EPS.
La commande `\image` traite aussi bien les images PNG que EPS, à ceci près que la commande `\image` insère un fond blanc, contrairement à `\epsfile`.
- La nouvelle commande de *markup* `\qr-code` permet d'insérer un QR-code de la taille spécifiée pour l'URL correspondante. Ceci peut servir à fournir un lien vers le site du compositeur ou de l'éditeur, ou bien vers les sources LilyPond ou des enregistrements, etc.