

LilyPond

Le système de gravure musicale

Utilisation des programmes

L'équipe de développement de LilyPond

Ce document constitue le manuel d'utilisation des programmes de GNU LilyPond 2.13.27. De plus, ce manuel suggère des « bonnes pratiques » pour une utilisation plus efficace.

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 <http://www.lilypond.org/>.

Copyright © 1999–2010 par les auteurs.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

La traduction de la notice de droits d'auteur ci-dessous vise à faciliter sa compréhension par le lecteur non anglophone, mais seule la notice en anglais a valeur légale.

Vous avez le droit de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU de documentation libre, version 1.1 ou tout autre version ultérieure publiée par la Free Software Foundation, “sans aucune section invariante”. Une copie de la licence est fournie à la section “Licence GNU de documentation libre”.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Pour LilyPond version 2.13.27

Table des matières

1	Exécution de lilypond	1
1.1	Utilisation habituelle	1
1.2	Utilisation en ligne de commande	1
	Lancement de <code>lilypond</code>	1
	Options en ligne de commande pour <code>lilypond</code>	1
	Variables d'environnement	1
	Exécution de LilyPond en mode protégé	1
1.3	Messages d'erreur	1
1.4	Quelques erreurs des plus courantes	1
	La musique déborde de la page	1
	Apparition d'une portée supplémentaire	1
	Erreur renvoyant à <code>../ly/init.ly</code>	3
	Message d'erreur «Unbound variable % »	3
	Message d'erreur «FT_Get_Glyph_Name »	3
2	Mise à jour avec <code>convert-ly</code>	4
2.1	LilyPond, une langue vivante	4
2.2	Exécution de <code>convert-ly</code>	4
2.3	Options en ligne de commande pour <code>convert-ly</code>	5
2.4	Problèmes d'exécution de <code>convert-ly</code>	5
2.5	Conversions manuelles	6
3	<code>lilypond-book</code> : association musique-texte	8
3.1	Exemple de document musicologique	8
3.2	Association musique-texte	8
	3.2.1 <code>L^AT_EX</code>	8
	3.2.2 <code>Texinfo</code>	8
	3.2.3 <code>HTML</code>	8
	3.2.4 <code>DocBook</code>	8
3.3	Options applicables aux fragments de musique	8
3.4	Utilisation de <code>lilypond-book</code>	8
3.5	Extensions de nom de fichier	8
3.6	Autres méthodes d'association texte-musique	8
	3.6.1 Extraction de plusieurs fragments d'une grande partition	8
	3.6.2 Insertion du résultat de LilyPond dans OpenOffice.org	8
	3.6.3 Insertion du résultat de LilyPond dans d'autres programmes	8
4	Programmes externes	9
4.1	Pointer-cliquer	9
4.2	LilyPond et les éditeurs de texte	10
	Mode Emacs	10
	Mode Vim	10
	Autres éditeurs	10
4.3	Conversion à partir d'autres formats	10
	4.3.1 Utilisation de <code>midi2ly</code>	11
	4.3.2 Utilisation de <code>musicxml2ly</code>	12
	4.3.3 Utilisation de <code>abc2ly</code>	13

4.3.4	Utilisation de <code>etf2ly</code>	14
4.3.5	Autres formats	14
4.4	Insertion de partition LilyPond dans d'autres programmes	14
	Extraction de plusieurs fragments d'une grande partition	14
	Insertion du résultat de LilyPond dans OpenOffice.org	14
	Insertion du résultat de LilyPond dans d'autres programmes	14
4.5	Inclusion du travail des autres	15
4.5.1	MIDI et articulations	15
5	Suggestions pour la saisie de fichiers	16
5.1	Suggestions générales	16
5.2	Gravure de musique existante	17
5.3	Projets d'envergure	17
5.4	Résolution de problèmes	18
5.5	De la commande <code>make</code> et des fichiers <code>Makefile</code>	18
Annexe A	GNU Free Documentation License	25
Annexe B	Index de LilyPond	32

1 Exécution de lilypond

1.1 Utilisation habituelle

1.2 Utilisation en ligne de commande

Lancement de lilypond

Options en ligne de commande pour lilypond

Variables d'environnement

Exécution de LilyPond en mode protégé

1.3 Messages d'erreur

1.4 Quelques erreurs des plus courantes

Les conditions amenant aux erreurs qui suivent sont fréquentes, bien qu'elles ne soient pas évidentes ni facilement localisables. Nous espérons que ces explications vous aideront à les résoudre plus facilement.

La musique déborde de la page

Lorsque la musique s'épanche au delà de la marge droite ou bien semble anormalement comprimée, la raison en est le plus souvent une note à la durée erronée ; cela finit par provoquer le débordement de la dernière note d'une mesure. Rien ne s'oppose à ce que la dernière note d'une mesure ne s'arrête avant la barre de mesure ; on considère simplement qu'elle se prolonge sur la mesure suivante. Des débordements à répétition finissent par générer une musique comprimée ou qui sort de la page, pour la simple et bonne raison que les sauts de ligne automatiques ne peuvent intervenir qu'à la fin d'une mesure complète, autrement dit lorsque toutes les notes sont terminées avant la fin de la mesure.

Note : Une durée erronée peut empêcher les sauts de ligne, ce qui conduit à une musique compressée, voire à un débordement de la page.

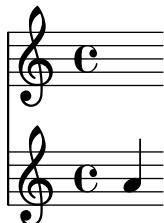
Une erreur de durée sera bien plus facilement localisable si vous positionnez régulièrement des contrôles de barre de mesure – voir [Section “Vérification des limites et numéros de mesure”](#) dans *Manuel de notation*.

Si vous tenez absolument à enchaîner de tels débordements, vous devrez insérer des barres de mesure invisibles là où vous souhaitez positionner un saut de ligne. Consultez le chapitre [Section “Barres de mesure”](#) dans *Manuel de notation* pour plus de détails.

Apparition d'une portée supplémentaire

Lorsque les contextes ne sont pas créés explicitement par la commande `\new`, ils le seront si la commande à exécuter n'est pas censée s'appliquer au contexte en cours. Pour des partitions simples, le fait que les contextes soient automatiquement créés rend bien des services, et c'est d'ailleurs le cas pour la majorité des exemples contenus dans les manuels de LilyPond. Cependant, la création implicite d'un contexte peut aboutir à l'apparition d'une portée « parasite ». On s'attend par exemple, en lisant le code qui suit, à ce que toutes les têtes de notes soient en rouge, alors que le résultat nous présente deux portées et que les notes, placées sur la portée inférieure, restent en noir.

```
\override Staff.NoteHead #'color = #red
\new Staff { a }
```



Étant donné qu'aucun contexte **Staff** n'existe lorsque la dérogation est introduite, LilyPond le crée implicitement pour lui appliquer la directive considérée. Survient alors la commande `\new Staff` qui, à son tour, crée une nouvelle portée pour contenir les notes qui suivent. Voici la syntaxe correcte pour obtenir ces notes en rouge :

```
\new Staff {
  \override Staff.NoteHead #'color = #red
  a
}
```



Autre exemple : la présence d'une commande `\relative` à l'intérieur d'une section `\repeat` générera obligatoirement une portée intempestive. Cela tient au fait que la commande `\repeat` va créer deux blocs `\relative` qui, chacun à leur tour, créeront implicitement un bloc **Staff** assorti d'un bloc **Voice**.

```
\repeat unfold 2 \relative { c d e f }
```



La manière adéquate de procéder consiste à inverser les commandes `\repeat` et `\relative`, comme ceci :

```
\relative {
  \repeat unfold 2 { c d e f }
}
```



Erreur renvoyant à ../ly/init.ly

Certains messages d'erreur relatifs à une erreur de syntaxe dans le fichier `../ly/init.ly` peuvent survenir lorsque le fichier est mal formaté. Cela se produit notamment lors d'un défaut de parité de bornages ou de guillemets.

L'erreur la plus courante est la simple omission d'une accolade fermante (`}`) à la fin du bloc `Score`. La solution est évidente en pareil cas : il suffit de vérifier que le bloc `Score` est bien clôturé. La structure des fichiers LilyPond est abordée plus en détails au chapitre [\[Organisation des fichiers LilyPond\]](#), page [\[undefined\]](#). C'est la raison pour laquelle nous vous invitons à utiliser un éditeur de texte qui prenne en charge le contrôle de parité des parenthèses, crochets et accolades afin de vous éviter de telles erreurs.

Lorsqu'il s'agit d'un guillemet fermant (`"`) omis, le message d'erreur devrait vous indiquer un numéro de ligne avoisinant. L'erreur se situe la plupart du temps une ou deux lignes au-dessus de celle indiquée.

Message d'erreur « Unbound variable % »

Ce message d'erreur, qu'il apparaisse sur le terminal ou en fin de fichier journal, est associé à un message du type « `GUILE` a signalé une erreur . . . », survient à chaque fois qu'un commentaire *LilyPond* est indûment placé dans une routine *Scheme*.

Un commentaire LilyPond est introduit par le signe pourcent (`%`) et ne doit en aucun cas se trouver dans une routine *Scheme*. En *Scheme*, les commentaires s'introduisent par un point-virgule (`;`).

Message d'erreur « FT_Get_Glyph_Name »

Ce message d'erreur, qu'il apparaisse sur le terminal ou en fin de fichier journal, survient lorsqu'un fichier source contient des caractères non ASCII et qu'il n'a pas été enregistré avec un encodage UTF-8. Pour plus de détails, reportez-vous au chapitre [Section "Codage du texte"](#) dans *Manuel de notation*.

2 Mise à jour avec `convert-ly`

La syntaxe des fichiers LilyPond évolue avec le temps, que ce soit dans le but de la simplifier ou pour l'améliorer. Il en résulte que l'interpréteur de LilyPond se retrouve incapable de traiter d'anciens fichiers. L'utilitaire `convert-ly` permet cependant de mettre ces fichiers en conformité au fur et à mesure que de nouvelles versions de LilyPond sont disponibles.

2.1 LilyPond, une langue vivante

La syntaxe de LilyPond change de temps en temps. Ces changements de syntaxe (le langage d'entrée) accompagnent les améliorations du logiciel. Ces changements sont parfois destinés à rendre les fichiers plus faciles à lire et à écrire, ou permettent d'intégrer de nouvelles fonctionnalités.

Par exemple, tous les noms des propriétés de `\paper` et `\layout` devaient être étiquetés sous la forme `premier-deuxième-troisième`. Nous avons constaté, une fois la version 2.11.60 mise à disposition, que la propriété `printallheaders` ne respectait pas cette convention. Aurions-nous dû la laisser telle que, au risque de dérouter les nouveaux utilisateurs par cette exception de formatage, ou bien la modifier – ce qui allait obliger ceux qui l'avaient déjà utilisée à se mettre en chasse ? Pour ce cas d'espèce, nous avons décidé de changer pour `print-all-headers`. Cette modification peut heureusement être automatisée par notre utilitaire `convert-ly`.

Malheureusement, `convert-ly` ne peut pas réaliser toutes les modifications. Par exemple, dans les versions 2.4 et antérieures de LilyPond, les accents et les lettres non anglaises étaient entrées en utilisant LaTeX – par exemple, `No\`e1`. À partir de la version 2.6, le caractère `ë` doit être entré directement dans le fichier LilyPond comme caractère UTF-8. `convert-ly` ne peut pas changer tous les caractères LaTeX en caractères UTF-8 ; vous devez mettre à jour vos vieux fichiers LilyPond manuellement.

2.2 Exécution de `convert-ly`

`convert-ly` utilise les mentions de `\version` – que vous n'avez sûrement pas oublié de porter dans vos fichiers – pour déterminer le numéro de l'ancienne version. Mettre à jour votre fichier ne vous demande que de lancer

```
convert-ly -e monfichier.ly
```

dans le dossier où il se trouve. `monfichier.ly` sera mis à jour, et vous aurez une copie de l'original : `monfichier.ly~`.

Note : `convert-ly` effectuera les conversions jusqu'aux modifications de syntaxe les plus récentes qu'il contient. C'est la raison pour laquelle le numéro de `\version` modifié est la plupart du temps inférieur au propre numéro de version de `convert-ly`.

Vous pouvez convertir tous les fichiers d'un dossier en lançant

```
convert-ly -e *.ly
```

Vous pouvez aussi affecter un autre nom au fichier mis à jour et conserver votre fichier original en l'état :

```
convert-ly monfichier.ly > monnouveau fichier.ly
```

Le programme affichera les numéros de version correspondant aux différentes conversions effectuées. Si aucun numéro de version n'apparaît, considérez que le fichier ne comporte pas de syntaxe obsolète.

Les utilisateurs de MacOS X disposent d'une entrée spécifique dans le menu : **Compile > Update syntax**.

Si vous utilisez windows, ouvrez un interpréteur de commande en faisant `Démarrer > Accessoires > Interpréteur de commandes`.

2.3 Options en ligne de commande pour `convert-ly`

L'utilitaire `convert-ly` se lance de la manière suivante :

```
convert-ly [option]... fichier...
```

Vous pouvez utiliser les options :

- `-e, --edit`
pour éditer directement le fichier d'origine.
- `-f, --from=from-patchlevel`
pour définir le numéro de version à partir duquel vous voulez effectuer les conversions. Lorsque cette option n'est pas activée, `convert-ly` tentera de le déterminer sur la foi de la mention de `\version` contenue dans le fichier. Cette option s'utilise sous la forme : `--from=2.10.25`
- `-n, --no-version`
Normalement, `convert-ly` ajoutera une indication de `\version` à votre fichier s'il n'en comporte pas. Cette option permet de passer outre.
- `-s, --show-rules`
pour afficher les conversions applicables.
- `--to=to-patchlevel`
pour n'appliquer les conversions que jusqu'à une version déterminée. Il s'agit par défaut de la dernière version disponible. Cette option s'utilise sous la forme : `--to=2.12.2`
- `-h, --help`
visualiser l'aide et quitter.

Lorsqu'il s'agit de fragments inclus dans un fichier texinfo, il vous faudra lancer

```
convert-ly --from=... --to=... --no-version *.itely
```

Lorsque vous désirez savoir quels changements de syntaxe sont intervenus entre deux versions de LilyPond, lancez

```
convert-ly --from=ancienne --to=récente -s
```

2.4 Problèmes d'exécution de `convert-ly`

Sous Windows, lorsque le nom du fichier original ou le chemin qui y mène comporte des espaces, l'interpréteur de commande requiert qu'il soit entouré de triples guillemets comme ci-dessous :

```
convert-ly ""D:/Mes Partitions/Ode.ly"" > "D:/Mes Partitions/nouveau Ode.ly"■
```

Lorsque la commande `convert-ly -e *.ly` échoue parce que son expansion dépasse la taille maximale d'une ligne, vous pouvez lancer `convert-ly` dans une boucle. L'exemple suivant permet, sous Unix, de convertir tous les fichiers `.ly` d'un même répertoire :

```
for f in *.ly; do convert-ly -e $f; done;
```

Avec l'interpréteur de commandes de Windows, la syntaxe consacrée est :

```
for %x in (*.ly) do convert-ly -e "%x"
```

Toutes les évolutions du langage ne sont pas forcément prises en charge. `convert-ly` ne tolère qu'une seule option de sortie à la fois. La mise à jour automatique du code Scheme inclus dans les fichiers LilyPond est plus qu'hasardeuse ; attendez-vous à devoir mettre les mains dans le cambouis.

2.5 Conversions manuelles

En théorie, un programme tel que `convert-ly` devrait pouvoir traiter n'importe quel changement de syntaxe. En effet, si un programme informatique sait interpréter aussi bien une version que l'autre, un autre programme informatique doit alors être capable de traduire un fichier donné¹.

Le projet LilyPond ne dispose cependant que de ressources limitées : les conversions ne sont pas toutes automatisées. Voici une liste de problèmes clairement identifiés :

1.6->2.0:

Doesn't always convert figured bass correctly, specifically things like {< >}. Mats' comment on working around this:

To be able to run `convert-ly`

on it, I first replaced all occurrences of '{<' to some dummy like '{#' and similarly I replaced '>}' with '&}'. After the conversion, I could then change back from '{ #' to '{ <' and from '& }' to '> }'.

Doesn't convert all text markup correctly. In the old markup syntax, it was possible to group a number of markup commands together within parentheses, e.g.

```
-#((bold italic) "string")
```

This will incorrectly be converted into

```
-\markup{{\bold italic} "string"}
```

instead of the correct

```
-\markup{\bold \italic "string"}
```

2.0->2.2:

Doesn't handle `\partcombine`

Doesn't do `\addlyrics => \lyricsto`, this breaks some scores with multiple stanzas.

2.0->2.4:

`\magnify` isn't changed to `\fontsize`.

```
- \magnify #m => \fontsize #f, where f = 6ln(m)/ln(2)
```

`remove-tag` isn't changed.

```
- \applyMusic #(remove-tag '. . .) => \keepWithTag #' . . .
```

`first-page-number` isn't changed.

```
- first-page-number no => print-first-page-number = ##f
```

Line breaks in header strings aren't converted.

```
- \\\ as line break in \header strings => \markup \center-align <
  "First Line" "Second Line" >
```

Crescendo and decrescendo terminators aren't converted.

```
- \rced => \!
```

```
- \rc => \!
```

2.2->2.4:

`\turnOff` (used in `\set Staff.VoltaBracket = \turnOff`) is not properly converted.

2.4.2->2.5.9

`\markup{ \center-align <{ ... }> }` should be converted to:

```
\markup{ \center-align {\line { ... }} }
```

but now, `\line` is missing.

2.4->2.6

Special LaTeX characters such as $\$~\$$ in text are not converted to UTF8.

2.8

¹ At least, this is possible in any LilyPond file which does not contain scheme. If there is scheme in the file, then the LilyPond file contains a Turing-complete language, and we run into problems with the famous « Halting Problem » in computer science.

`\score{}` must now begin with a music expression. Anything else (particularly `\header{}`) must come after the music.

3 lilypond-book : association musique-texte

3.1 Exemple de document musicologique

Fichier d'entrée

Traitement

Résultat

3.2 Association musique-texte

3.2.1 \LaTeX

3.2.2 Texinfo

3.2.3 HTML

3.2.4 DocBook

Conventions communes

Inclusion d'un fichier LilyPond

Inclusion de code LilyPond

Génération du document DocBook

3.3 Options applicables aux fragments de musique

3.4 Utilisation de lilypond-book

Instructions spécifiques à certains formats

\LaTeX

Texinfo

Options en ligne de commande

3.5 Extensions de nom de fichier

3.6 Autres méthodes d'association texte-musique

3.6.1 Extraction de plusieurs fragments d'une grande partition

3.6.2 Insertion du résultat de LilyPond dans OpenOffice.org

3.6.3 Insertion du résultat de LilyPond dans d'autres programmes

4 Programmes externes

LilyPond peut interagir avec d'autres programmes, selon différentes manières.

4.1 Pointer-cliquer

Le pointer-cliquer (*point and click*) permet de se retrouver directement dans le fichier source, à la note que l'on pointe dans le visionneur de PDF. Ceci facilite grandement le repérage des erreurs à partir du fichier imprimable.

Lorsque cette fonctionnalité est active, LilyPond ajoute des hyperliens au fichier PDF. Ces liens sont transmis au navigateur internet qui se charge d'ouvrir un éditeur de texte à l'endroit même où le curseur pointe.

Afin que cette chaîne de traitement soit pleinement opérationnelle, il faut configurer votre visionneur de PDF de façon à ce qu'il suive les liens grâce au script 'lilypond-invoke-editor' fourni avec LilyPond.

Pour Xpdf, sous UNIX, vous devrez insérer la ligne suivante dans le fichier 'xpdfrc' – soit '/etc/xpdfrc', soit dans votre répertoire personnel '.xpdfrc'.

```
urlCommand      "lilypond-invoke-editor %s"
```

'lilypond-invoke-editor' est un petit programme assistant. Il se charge d'appeler un éditeur pour les identifiants de ressource (*URI*) de type `textedit`, et un navigateur pour les autres. Il teste en outre la variable d'environnement `EDITOR` pour les cas suivant :

```
emacs      sera alors lancée la commande
            emacsclient --no-wait +line:column file
gvim       sera alors lancée la commande
            gvim --remote +:line:normcolumn file
nedit      sera alors lancée la commande
            nc -noask +line file'
```

La variable d'environnement `LYEDITOR` permet d'anticiper cette affectation, puisqu'elle contient la commande qui permet de lancer l'éditeur en tenant compte des valeurs respectives de `%(file)s`, `%(column)s`, et `%(line)s`. Par exemple,

```
emacsclient --no-wait +% (line)s: %(column)s %(file)s
```

en variable d'environnement `LYEDITOR` revient au lancement d'un client emacs standard.

L'option pointer-cliquer accroît la taille des fichiers de manière significative. Afin de réduire la taille des fichiers PDF et PS, il est toujours possible de désactiver le pointer-cliquer en ajoutant

```
\pointAndClickOff
```

dans le fichier '.ly'. Il peut alors être activé de manière explicite grâce à

```
\pointAndClickOn
```

Le pointer-cliquer peut aussi être désactivé au moment de la compilation en ligne de commande :

```
lilypond -dno-point-and-click file.ly
```

Note : Lorsqu'un fichier LilyPond est destiné à être redistribué, pensez à désactiver le pointer-cliquer, de telle sorte que les chemins d'accès et autres informations propres à votre système ne se retrouvent pas inclus dans le fichier '.pdf'.

4.2 LilyPond et les éditeurs de texte

Certains éditeurs de texte prennent en charge LilyPond.

Mode Emacs

Emacs dispose d'un 'lilypond-mode' qui fournit l'autocomplétion des mots-clés, l'indentation, les appariements spécifiques à LilyPond, la coloration syntaxique, ainsi que des raccourcis pour compiler et consulter les manuels de LilyPond en mode info. Si le 'lilypond-mode' n'était pas installé sur votre système, procédez comme ci-dessous.

Le répertoire 'elisp' inclus dans les sources contient aussi un mode pour saisir la musique et lancer LilyPond. Faites `make install` pour l'installer dans votre *elispdir*. Le fichier 'lilypond-init.el' devrait trouver sa place dans *load-path*'/site-start.d/' ou bien ajouté à votre '~/.emacs' ou '~/.emacs.el'.

En tant que simple utilisateur, vous pouvez toujours ajouter votre propre répertoire (par exemple '~/.site-lisp/') à votre *load-path* en ajoutant la ligne suivante – modifiée en conséquence – à votre '~/.emacs' :

```
(setq load-path (append (list (expand-file-name "~/site-lisp")) load-path))
```

Mode Vim

En ce qui concerne **VIM**, LilyPond fournit un fichier 'vimrc' qui gère la coloration syntaxique. Le répertoire \$VIM inclus dans les sources contient aussi un mode pour saisir la musique et lancer LilyPond.

Le type de fichier LilyPond sera reconnu si votre '~/.vim/filetype.vim' contient

```
if exists("did_load_filetypes")
  finish
endif
augroup filetypedetect
  au! BufNewFile,BufRead *.ly,*.ily          setf lilypond
augroup END
```

N'oubliez pas d'inclure ce chemin en ajoutant à votre '~/.vimrc' la ligne suivante :

```
set runtimepath+="/usr/local/share/lilypond/${LILYPOND_VERSION}/vim/
```

où \${LILYPOND_VERSION} correspond à votre version de LilyPond. Si LilyPond n'est pas installé dans '/usr/local/', modifiez ce chemin en conséquence. À titre d'exemple, si vous avez une Fedora, le chemin mène à la version de Vim au lieu de LilyPond :

```
set runtimepath+="/usr/share/vim/vim72/
```

Autres éditeurs

LilyPond est pris en charge par d'autres éditeurs, aussi bien en mode texte qu'avec une interface graphique. Dans la mesure où leurs fichiers de configuration ne sont pas fournis avec LilyPond, nous vous invitons à consulter leur documentation pour plus d'information. Certains de ces éditeurs sont mentionnés à la page [Section "Facilités d'édition" dans Informations générales](#).

4.3 Conversion à partir d'autres formats

La musique peut aussi être récupérée par importation d'un autre format. Ce chapitre passe en revue les différents outils prévus à cet effet et inclus dans la distribution. Il existe d'autres outils qui permettent de générer du code LilyPond, comme par exemple des séquenceurs en mode graphique ou des convertisseurs XML. Pour plus de détails, rendez-vous sur le [site](#).

Il s'agit de programmes distincts de `lilypond` qui se lancent en ligne de commande. Pour plus de précisions, reportez-vous au chapitre [Section 1.2 \[Utilisation en ligne de commande\]](#), page 1.

Si vous utilisez MacOS 10.3 ou 10.4 et recontrez quelque problème avec l'un de ces scripts, comme `convert-ly`, reportez-vous à la page [Section “MacOS X” dans Informations générales](#).

Problèmes connus et avertissements

Les développeurs ne sont malheureusement pas suffisamment nombreux et disponibles pour maintenir à jour ces programmes, considérez-les donc *en l'état*. Nous acceptons les patches avec plaisir, mais il y a peu de chance pour que nous soyons en mesure de résoudre les bogues de ces programmes.

4.3.1 Utilisation de midi2ly

`midi2ly` traduit un fichier MIDI de Type 1 en un fichier source LilyPond.

MIDI (Music Instrument Digital Interface) constitue un standard pour les instruments. Il spécifie le câblage, un protocole série et un format de fichier. Le format de fichier MIDI est de ce fait un format standard pour exporter de la musique à partir d'autres programmes, et cette faculté prend tout son intérêt lorsqu'un programme dispose d'un convertisseur capable d'importer directement un fichier.

`midi2ly` convertit les pistes en contextes de portée ([Section “Staff” dans Référence des propriétés internes](#)) et les canaux en contextes de voix ([Section “Voice” dans Référence des propriétés internes](#)). Les hauteurs sont rendues en mode relatif, et les durées spécifiées lorsque nécessaire.

Vous pouvez enregistrer un fichier MIDI grâce à un clavier électronique et ensuite le convertir en fichier `.ly`. Néanmoins le rythme humain n'a pas une précision suffisante pour qu'une conversion MIDI à `ly` ne se fasse sans surprise. En le couplant à une quantisation (options `-s` et `-d`), `midi2ly` tente de compenser dans la mesure du possible ces problèmes de temporisation. C'est la raison pour laquelle le recours à `midi2ly` n'est pas recommandé pour des fichiers midi générés manuellement.

Pour le lancer en ligne de commande, procédez ainsi :

```
midi2ly [option]... fichier-midi
```

Notez bien que, par « ligne de commande », nous parlons de la ligne de commande du système. Pour plus de précisions, reportez-vous à [\(undefined\) \[Converting from other formats\], page \(undefined\)](#).

`midi2ly` accepte les options suivantes :

`-a, --absolute-pitches`

Rendu en hauteurs absolues.

`-d, --duration-quant=DUR`

Quantiser les durées à partir de *DUR*.

`-e, --explicit-durations`

Rendu explicite des durées.

`-h, --help`

Afficher un résumé des utilisations.

`-k, --key=acc[:minor]`

Déterminer la tonalité par défaut. *acc* > 0 fixe le nombre de dièses ; *acc* < 0 le nombre de bémols. Une tonalité mineure est spécifiée par l'emploi de `:1`.

`-o, --output=file`

Générer le résultat dans le fichier *file*.

`-s, --start-quant=DUR`

Quantiser le début des notes sur *DUR*.

`-t, --allow-tuplet=DUR*NUM/DEN`
 Accepter des n-olets de valeur *DUR*NUM/DEN*.

`-v, --verbose`
 Mode verbeux.

`-V, --version`
 Afficher le numéro de version.

`-w, --warranty`
 Afficher les mentions de garantie et de copyright.

`-x, --text-lyrics`
 Interpréter le texte comme des paroles.

Problèmes connus et avertissements

Le tuilage en arpège ne sera pas rendu correctement. La première note sera lue et les suivantes ignorées. Affectez-leur une durée unique et ajoutez une indication de phrasé ou de pédale.

4.3.2 Utilisation de musicxml2ly

MusicXML est un dialecte XML pour représenter la notation musicale.

`musicxml2ly` extrait, à partir d'un fichier MusicXML, les notes, articulations, structure de la partition, paroles, etc. et les écrit dans un fichier *.ly*. Il se lance en ligne de commande.

Pour le lancer en ligne de commande, procédez ainsi :

```
musicxml2ly [option]... fichier-xml
```

Notez bien que, par « ligne de commande », nous parlons de la ligne de commande du système. Pour plus de précisions, reportez-vous à [\[Converting from other formats\]](#), page [\[undefined\]](#).

Si le nom du fichier est '-', `musicxml2ly` lira directement à partir de la ligne de commande.

`musicxml2ly` accepte les options suivantes :

`-a, --absolute`
 Rendu en hauteurs absolues.

`-h, --help`
 Afficher un résumé des utilisations.

`-l, --language=LANG`
 Utiliser un autre fichier de définition linguistique '*LANG.ly*' et les noms de notes correspondants, comme par exemple *deutsch* pour '*deutsch.ly*' et noms de notes en allemand.

`--lxml`
 Utiliser le packaging Python `lxml.etree`, moins gourmand en mémoire et temps de calcul, pour effectuer l'analyse XML.

`--nd --no-articulation-directions`
 Ne pas convertir la direction (^, _ ou -) des articulations, nuances, etc.

`--no-beaming`
 Ne pas convertir les informations de ligature ; laisser LilyPond gérer les ligatures automatiquement.

`-o, --output=file`
 Générer le résultat dans le fichier *fichier*. S'il n'est pas déterminé, ce sera *fichier-xml'.ly'* ; '-' produira le résultat sur la sortie standard (*stdout*).

`-r,--relative`
Rendu en hauteurs relatives (mode par défaut).

`-v,--verbose`
Mode verbeux.

`--version`
Afficher le numéro de version.

`-z,--compressed`
Le fichier d'entrée est un fichier MusicXML zippé.

4.3.3 Utilisation de abc2ly

Note : Ce programme ne bénéficie d'aucune maintenance. Il est susceptible d'être supprimé des versions futures de LilyPond.

ABC est un format relativement simple basé sur l'ASCII. Sa description est disponible sur le [site d'ABC](#).

abc2ly traduit du format ABC au format LilyPond.

Pour le lancer en ligne de commande, procédez ainsi :

```
abc2ly [option]... fichier-abc
```

abc2ly accepte les options suivantes :

`-b,--beams=None`
Préserver la notion de lien de croches propre à ABC.

`-h,--help`
Afficher un résumé des utilisations.

`-o,--output=file`
Générer le résultat dans le fichier *file*.

`-s,--strict`
Être strict sur la réussite.

`--version`
Afficher le numéro de version.

Il est possible d'ajouter des bribes de code LilyPond dans un fichier source ABC. Ainsi, l'assertion

```
%%LY voices \set autoBeaming = ##f
```

aura pour conséquence d'insérer le texte qui suit le mot-clé « voices » dans la voix correspondante du fichier LilyPond.

De la même manière,

```
%%LY slyrics more words
```

placera le texte suivant le mot-clé « slyrics » dans une ligne de paroles.

Problèmes connus et avertissements

Le standard ABC n'est pas si « standard » que cela. Pour des fonctionnalités étendues, comme la polyphonie, existent différentes conventions.

Un fichier comportant plusieurs morceaux ne peut être converti.

ABC synchronise paroles et musique en début de ligne ; abc2ly ne le fait pas.

abc2ly ignore les ligatures ABC.

4.3.4 Utilisation de etf2ly

Note : Ce programme ne bénéficie d’aucune maintenance. Il est susceptible d’être supprimé des versions futures de LilyPond.

ETF (Enigma Transport Format) est l’un des formats utilisés par le logiciel Finale, édité par Coda Music Technology. **etf2ly** convertit partiellement les fichiers ETF en fichiers source LilyPond.

Pour le lancer en ligne de commande, procédez ainsi :

```
etf2ly [option]... fichier-etf
```

Notez bien que, par « ligne de commande », nous parlons de la ligne de commande du système. Pour plus de précisions, reportez-vous à [\[Converting from other formats\]](#), page [\[undefined\]](#).

etf2ly accepte les options suivantes :

-h, --help

Afficher cette aide.

-o, --output=*file*

Générer le résultat dans le fichier *file*.

--version

Afficher le numéro de version.

Problèmes connus et avertissements

La liste des scripts d’articulation est incomplète. Les mesures vides perturbent **etf2ly**. Les séquences de notes d’ornement ne se terminent pas de manière satisfaisante.

4.3.5 Autres formats

LilyPond ne prend pas en charge d’autre format. Cependant, certains outils indépendants permettent de générer des fichiers LilyPond, comme indiqué à la page [Section “Facilités d’édition”](#) dans *Informations générales*.

4.4 Insertion de partition LilyPond dans d’autres programmes

Nous allons nous intéresser ici à différents moyens pour associer texte et musique, en laissant de côté l’automatisation grâce à **lilypond-book**.

Extraction de plusieurs fragments d’une grande partition

Si d’aventure vous deviez recopier un certain nombre d’extraits d’une même partition, vous pouvez recourir à « l’emporte pièce » – la fonction *clip systems* – comme indiqué au chapitre [Section “Extraction de fragments musicaux”](#) dans *Manuel de notation*.

Insertion du résultat de LilyPond dans OpenOffice.org

OOoLilyPond permet d’insérer directement des partitions LilyPond dans OpenOffice.org.

Insertion du résultat de LilyPond dans d’autres programmes

Dans le cas où vous cherchez à insérer le résultat de LilyPond dans d’autres programmes, utilisez **lilypond** plutôt que **lilypond-book**. Chaque extrait devra être généré séparément avant d’être inséré dans votre document. De très nombreux programmes sont capables de contenir le résultat de LilyPond, que ce soit au format ‘PNG’, ‘EPS’, ou ‘PDF’.

Les options suivantes vous permettront de réduire notablement les contours de l’image LilyPond :

```
\paper{
  indent=0\mm
  line-width=120\mm
  oddFooterMarkup=##f
  oddHeaderMarkup=##f
  bookTitleMarkup = ##f
  scoreTitleMarkup = ##f
}
```

```
{ c1 }
```

En procédant comme ci-après, vous obtiendrez un fichier ‘EPS’ :

```
lilypond -dbackend=eps -dno-gs-load-fonts -dininclude-eps-fonts monfichier.ly
```

ou ‘PNG’ :

```
lilypond -dbackend=eps -dno-gs-load-fonts -dininclude-eps-fonts --png monfichier.ly■
```

4.5 Inclusion du travail des autres

Certains ont écrit des lignes et des lignes de code – souvent bien utiles – qui peuvent servir à différents projets. En attendant le jour où elles pourraient faire partie intégrante de LilyPond, vous pouvez toujours les télécharger et les utiliser avec la commande `\include`.

4.5.1 MIDI et articulations

LilyPond sait produire des fichiers MIDI, principalement dans le but de « contrôle-qualité » – heureux détenteurs d’une oreille absolue – de ce qui a été saisi. Ne seront toutefois reproduits, en plus des notes et durées, que les nuances et tempi explicites.

Le projet *articulate* (site en anglais) s’est donné pour objectif de reproduire plus d’informations dans le MIDI. Les notes qui ne sont pas liées sont ainsi raccourcies dans le but « d’articuler ». Ce raccourcissement dépend de l’articulation appliquée à la note : un *staccato* raccourcira la note de moitié, un *tenuto* lui gardera sa durée entière. . . Ce script réalise aussi les trilles et *grupettos* et compte bien traiter d’autres ornements tels que les mordants.

<http://www.nicta.com.au/people/chubbp/articulate>

Problèmes connus et avertissements

Ce projet ne peut traiter que ce qu’il connaît : tout ce qui peut ressembler à un *markup* – et donc pas à la propriété d’une note – sera ignoré.

5 Suggestions pour la saisie de fichiers

Maintenant vous êtes prêt à travailler sur de plus gros fichiers LilyPond – des pièces entières, et plus seulement les petits exemples du tutoriel. Mais comment devriez-vous vous y prendre ?

Tant que LilyPond parvient à comprendre vos fichiers et produit le résultat que vous souhaitez, peu importe la manière dont le code est organisé. Néanmoins, quelques critères doivent être pris en compte lorsque l'on écrit un fichier LilyPond.

- Si vous faites une erreur, la structure même du fichier LilyPond peut permettre de la localiser plus ou moins facilement.
- Et si vous souhaitez partager vos fichiers avec quelqu'un d'autre, ou si vous souhaitez modifier vos propres fichiers dans quelques années ? Si certains fichiers LilyPond sont compréhensibles au premier coup d'oeil, d'autres vous feront vous arracher les cheveux pendant une heure.
- Et si vous souhaitez mettre à jour votre fichier pour l'utiliser avec une version plus récente de LilyPond ? La syntaxe du langage d'entrée change parfois lorsque LilyPond s'améliore. La plupart des changements peuvent être appliqués automatiquement avec `convert-ly`, mais quelques-uns peuvent requérir une intervention manuelle. Vos fichiers LilyPond peuvent être structurés de manière à faciliter leur mise à jour.

5.1 Suggestions générales

Voici quelques conseils qui peuvent vous éviter certains problèmes ou en résoudre d'autres.

- **Ajoutez le numéro de version dans chaque fichier.** Notez que chaque fichier modèle contient une ligne `\version "2.13.27"`. Nous vous conseillons fortement d'inclure cette ligne, même pour de petits fichiers. Par expérience, il est très difficile de se rappeler quelle version de LilyPond on utilisait quelques années auparavant. L'utilitaire `convert-ly` demande que vous spécifiez la version de LilyPond vous utilisiez alors.
- **Ajoutez des contrôles :** *Section “Vérifications d’octave” dans Manuel de notation* et *Section “Vérification des limites et numéros de mesure” dans Manuel de notation*. Si vous avez ajouté des contrôles de loin en loin, et que vous faites une erreur, vous pourrez la retrouver plus rapidement. « De loin en loin », qu'est-ce à dire ? Cela dépend de la complexité de la musique. Pour de la musique très simple, peut-être une ou deux fois. Pour de la musique très complexe, peut-être à chaque mesure.
- **Une mesure par ligne de texte.** Si la musique en elle-même ou le résultat que vous désirez contient quelque chose de compliqué, il est souvent bon de n'écrire qu'une seule mesure par ligne. Économiser de la place en tassant huit mesures par ligne, ça ne vaut pas vraiment le coup si l'on doit corriger vos fichiers.
- **Ajoutez des commentaires.** Utilisez soit des numéros de mesure (assez souvent), soit des références au contenu musical – « second thème des violons », « quatrième variation », etc. Vous pouvez ne pas avoir besoin des commentaires lorsque vous écrivez une pièce pour la première fois, mais si vous souhaitez y revenir deux ou trois ans plus tard pour changer quelque chose, ou si vous donnez le fichier source à un ami, ce sera beaucoup plus difficile de déterminer vos intentions ou la manière dont votre fichier est structuré si vous n'y avez pas adjoint de commentaires.
- **Indentez les accolades.** Beaucoup de problèmes viennent d'un défaut de parité entre `{` et `}`.
- **Mentionnez les durées** au début de chaque section ou variable. Si vous saisissez `c4 d e` au début d'une phrase, vous vous épargnerez des problèmes si, plus tard, vous modifiez votre musique.
- **Séparez les affinages de mise en forme** de la musique elle-même. Voyez [\[Économie de saisie grâce aux identificateurs et fonctions\]](#), page [\[undefined\]](#) et [\[undefined\]](#) [\[Feuilles de style\]](#), page [\[undefined\]](#).

5.2 Gravure de musique existante

Si vous saisissez de la musique à partir d'une partition existante, c'est-à-dire de la musique déjà écrite,

- n'entrez qu'un seul système de la partition originale à la fois – mais toujours une seule mesure par ligne de texte –, et vérifiez chaque système lorsqu'il est terminé. Vous pouvez utiliser les commandes `showLastLength` et `showFirstLength` pour accélérer la compilation – voir [Section “Ignorer des passages de la partition”](#) dans *Manuel de notation* ;
- définissez `mBreak = {\break }` et insérez `\mBreak` dans le fichier d'entrée pour obtenir des sauts de ligne identiques à la partition originale. Cela facilite la comparaison entre la partition originale et la partition de LilyPond. Lorsque vous avez fini de relire votre musique, vous pouvez définir `mBreak = { }` pour enlever tous ces sauts de ligne, et laisser LilyPond placer les sauts de ligne selon son propre algorithme.
- encadrez les notes d'une partie pour instrument transpositeur dans un

```
\transpose c tonalité-naturelle {...}
```

(où *tonalité-naturelle* correspond à celle de l'instrument en question) de telle sorte que la musique comprise dans cette variable se retrouve en ut. Vous pourrez toujours transposer à l'inverse si besoin lorsque vous ferez appel à cette variable. Des erreurs de transposition seront moins susceptibles de se produire si la musique de toutes les variables est dans la même et unique tonalité.

De la même manière, prenez toujours le do comme note de départ ou d'arrivée. Ceci aura pour simple conséquence que les autres tonalités que vous utiliserez seront celles propres à chacun des instruments – sib pour une trompette en si bémol, ou lab pour une clarinette en la bémol.

5.3 Projets d'envergure

Lorsque l'on travaille sur un gros projet, il devient vital de structurer clairement ses fichiers LilyPond.

- **Utilisez un identificateur pour chaque voix**, avec un minimum de structure dans la définition. La structure de la section `\score` est la plus susceptible de changer, notamment dans une nouvelle version de LilyPond, alors que la définition du `violon` l'est beaucoup moins.

```
violin = \relative c'' {
  g4 c'8. e16
}
...
\score {
  \new GrandStaff {
    \new Staff {
      \violin
    }
  }
}
```

- **Séparez les retouches** des définitions de musique. Nous vous avons déjà invité à adopter une telle pratique, qui par ailleurs devient vitale pour des projets d'importance. Nous pouvons avoir besoin de changer la définition de `fpuisp`, mais dans ce cas nous n'aurons besoin de le faire qu'une seule fois, et nous pourrions encore éviter de modifier quoi que ce soit à l'intérieur de la définition du `violon`.

```
fpuisp = _\markup{
  \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p }
violin = \relative c'' {
```

```
g4\fpuisp c'8. e16
}
```

5.4 Résolution de problèmes

Tôt ou tard, vous écrirez un fichier que LilyPond ne peut pas compiler. Les messages que LilyPond affiche peuvent vous aider à trouver l'erreur, mais dans beaucoup de cas vous aurez besoin de faire quelques recherches pour déterminer la source du problème.

Pour ce faire, les outils les plus puissants sont le commentaire de fin de ligne, indiqué par %, et le commentaire multilignes (ou bloc de commentaire), indiqué par %{ ... %}. Si vous ne pouvez localiser le problème, commencez par mettre en commentaire de grandes parties de votre fichier source. Après avoir mis en commentaire une section, essayez de compiler à nouveau. Si cela fonctionne, c'est que le problème se situe dans cette partie du fichier. Si cela ne fonctionne pas, continuez à mettre en commentaire d'autres sections, jusqu'à ce que vous ayez quelque chose qui compile.

Dans un cas extrême, vous pourriez en arriver à

```
\score {
  <<
    % \melodie
    % \harmonie
    % \basse
  >>
  \layout{ }
}
```

c'est-à-dire un fichier sans aucune musique.

Si cela se produit, ne vous découragez pas. Décommentez un peu, la partie de basse par exemple, et voyez si ça fonctionne. Si ce n'est pas le cas, placez en commentaire toute la partie de basse, mais laissez `\basse` décommenté dans le bloc `\score`.

```
basse = \relative c' {
  %{
    c4 c c c
    d d d d
  %}
}
```

Maintenant commencez à décommenter petit à petit la partie de `basse` jusqu'à ce que vous localisiez la ligne qui pose problème.

Une autre technique de déboguage très utile est la construction d'[Section “Exemples minimaux” dans Informations générales](#).

5.5 De la commande make et des fichiers Makefile

La plupart des plates-formes sur lesquelles tourne LilyPond disposent d'un logiciel appelé `make`. Ce logiciel va lire un fichier spécial, nommé de `Makefile`, qui contient tout ce qu'il faut – les dépendances entre certains fichiers, les instructions successives à traiter par le système – pour aboutir au fichier que vous désirez obtenir. Il pourrait par exemple contenir tout ce qu'il faut pour produire `ballade.pdf` et `ballade.midi` à partir de `ballade.ly` en lançant LilyPond.

La création d'un `Makefile` peut se révéler pertinente pour certains projets, que ce soit par simple goût personnel ou bien par respect de ceux qui pourront accéder à vos sources. Cette manière de procéder est particulièrement indiquée lorsque vous travaillez sur un projet de grande envergure impliquant de nombreuses inclusions de fichiers et différentes éditions – par exemple un conducteur et un matériel d'orchestre complet avec la partition pour le chef et une partition

séparée pour chacun des pupitres – ou bien si votre projet requiert certaines commandes particulières comme `lilypond-book`. Les *Makefiles* varient tant en complexité qu'en flexibilité selon les besoin et les aptitudes de celui qui les crée. Le programme GNU Make est installé par défaut sur les distributions Linux et sur MacOS X, et il en existe une version pour les environnements Windows.

Consultez le **GNU Make Manual** pour plus de détails sur ce dont `make` est capable – vous pourrez même en trouver des versions françaises à l'aide des moteurs de recherche –, dans la mesure où ce qui suit ne donne qu'un bref aperçu de ses possibilités.

Les commandes permettant de définir les règles diffèrent selon la plate-forme : si les différents Linux et MacOS X utilisent `bash`, Windows utilise `cmd`. Dans le cas de MacOS X, vous devrez toutefois configurer votre système de telle sorte qu'il utilise l'interpréteur en ligne de commande. Voici quelques exemples de fichier *Makefile*, avec une version pour Linux ou MacOS et une pour Windows.

Pour commencer, une pièce à quatre mouvements pour orchestre et dont les fichiers sont répartis selon l'arborescence suivante :

```
Symphonie/
|-- MIDI/
|-- Makefile
|-- Notes/
|   |-- alto.ily
|   |-- cor.ily
|   |-- cello.ily
|   |-- figures.ily
|   |-- hautbois.ily
|   |-- trioCordes.ily
|   |-- violonOne.ily
|   `-- violonTwo.ily
|-- Partitions/
|   |-- symphonie.ly
|   |-- symphonieI.ly
|   |-- symphonieII.ly
|   |-- symphonieIII.ly
|   `-- symphonieIV.ly
|-- PDF/
|-- Pupitres/
|   |-- symphon-alto.ly
|   |-- symphonie-cello.ly
|   |-- symphonie-cor.ly
|   |-- symphonie-hautbois.ly
|   |-- symphonie-violonUn.ly
|   `-- symphonie-violonDeux.ly
`-- symphonieDefs.ily
```

Les fichiers `.ly` des répertoires `Partitions` et `Pupitres` récupéreront la notation des fichiers `.ily` contenus dans le répertoire `Notes` :

```
%% début du fichier "symphone-cello.ly"
\include ../symphonieDefs.ily
\include ../Notes/cello.ily
```

Le *Makefile* répertorie des cibles correspondant à `score` (l'intégrale au format conducteur), `mouvements` (chacun des mouvements au format conducteur) et `pupitres` (une partition par pupitre). Il contient aussi une cible `archive` chargée de générer une archive des fichiers source

qui pourra être diffusée sur la toile ou transmise par courriel. Voici ce que contiendrait ce *Makefile* pour Linux ou MacOS X. Ce fichier doit être enregistré sous le nom de *Makefile* à la racine du projet – ici *Symphonie*.

Note : Lorsque vous définissez une cible ou une règle sur plusieurs lignes, les lignes à partir de la deuxième **doivent** débiter par une tabulation, non pas par des espaces.

```
# Le préfixe au nom des fichiers résultants
piece = symphonie
# Détermination du nombre de processeurs
CPU_CORES=`cat /proc/cpuinfo | grep -ml "cpu cores" | sed s/".*: "//`
# La commande d'appel à lilypond
LILY_CMD = lilypond -ddelete-intermediate-files \
            -dno-point-and-click -djob-count=$(CPU_CORES)

# Les suffixes utilisés dans ce Makefile
.SUFFIXES: .ly .ily .pdf .midi

# Les fichiers sources et résultants sont recherchés dans les répertoires
# listés dans la variable VPATH. Ceux-ci sont tous des sous-répertoires
# du répertoire courant (fourni par la variable de GNU make `CURDIR').
VPATH = \
    $(CURDIR)/Partitions \
    $(CURDIR)/PDF \
    $(CURDIR)/Pupitres \
    $(CURDIR)/Notes

# La règle type pour créer un PDF et un MIDI à partir d'un fichier
# source LY.
# Les .pdf résultants iront dans le sous-répertoire "PDF" et les fichiers
# .midi dans le sous-répertoire "MIDI".
%.pdf %.midi: %.ly
    $(LILY_CMD) $<; \           # cette ligne commence par une tabulation
    if test -f "$*.pdf"; then \
        mv "$*.pdf" PDF/; \
    fi; \
    if test -f "$*.midi"; then \
        mv "$*.midi" MIDI/; \
    fi

notes = \
    alto.ily \
    cello.ily \
    cor.ily \
    hautbois.ily \
    violonUn.ily \
    violonDeux.ily

# Les dépendances selon le mouvement.
$(piece)I.pdf: $(piece)I.ly $(notes)
$(piece)II.pdf: $(piece)II.ly $(notes)
```

```

$(piece)III.pdf: $(piece)III.ly $(notes)
$(piece)IV.pdf: $(piece)IV.ly $(notes)

# Les dépendances pour la partition intégrale.
$(piece).pdf: $(piece).ly $(notes)

# Les dépendances pour les pupitres.
$(piece)-alto.pdf: $(piece)-alto.ly alto.ily
$(piece)-cello.pdf: $(piece)-cello.ly cello.ily
$(piece)-cor.pdf: $(piece)-cor.ly cor.ily
$(piece)-hautbois.pdf: $(piece)-hautbois.ly hautbois.ily
$(piece)-violonUn.pdf: $(piece)-violonUn.ly violonUn.ily
$(piece)-violonDeux.pdf: $(piece)-violonDeux.ly violonDeux.ily

# Lancer `make score' pour générer l'intégrale des quatre mouvements en
# un seul fichier.
.PHONY: score
score: $(piece).pdf

# Lancer `make parties' pour obtenir tous les pupitres.
# Lancer `make toto.pdf' pour obtenir la partie instrumentale de toto.
# Par exemple : `make symphonie-cello.pdf'.
.PHONY: parties
parties: $(piece)-cello.pdf \
        $(piece)-violonUn.pdf \
        $(piece)-violonDeux.pdf \
        $(piece)-alto.pdf \
        $(piece)-hautbois.pdf \
        $(piece)-cor.pdf

# Lancer `make mouvements' pour générer un fichier séparé pour chacun
# des mouvements.
.PHONY: mouvements
mouvements: $(piece)I.pdf \
            $(piece)II.pdf \
            $(piece)III.pdf \
            $(piece)IV.pdf

all: score parties mouvements

archive:
    tar -cvvf symphonie.tar \      # cette ligne commence par une tabulation
    --exclude=*pdf --exclude=*~ \
    --exclude=*midi --exclude=*.tar \
    ../Symphonie/*

```

Les choses se compliquent sous Windows. Une fois GNU Make pour Windows téléchargé et installé, il vous faudra correctement définir le chemin d'accès au programme *Make* – dans les variables d'environnement du système – afin que l'interpréteur de commandes DOS puisse le localiser. Pour cela, faites un clic droit sur « Poste de travail », choisissez **Propriétés** puis **Avancées**. Cliquez sur **Variables d'environnement** puis, dans l'onglet **Variables système**,

mettez **path** en surbrillance et cliquez sur **Modifier**. Ajoutez alors le chemin d'accès complet à l'exécutable de GNU Make, qui devrait ressembler à :

```
C:\Program Files\GnuWin32\bin
```

Il va également falloir adapter le *makefile* aux particularités de l'interpréteur de commandes et à la présence d'espaces dans le nom de certains répertoire de ce système. La cible **archive** est tout bonnement supprimée, puisque Windows ne dispose pas de la commande **tar**. Enfin, les fichiers MIDI ont une extension par défaut propre à Windows.

```
## VERSION POUR WINDOWS
##
piece = symphonie
LILY_CMD = lilypond -ddelete-intermediate-files \
               -dno-point-and-click \
               -djob-count=$(NUMBER_OF_PROCESSORS)

#get the 8.3 name of CURDIR (workaround for spaces in PATH)
workdir = $(shell for /f "tokens=*" %%b in ("$(CURDIR)") \
do @echo %%~sb)

.SUFFIXES: .ly .ily .pdf .mid

VPATH = \
    $(workdir)/Partitions \
    $(workdir)/PDF \
    $(workdir)/Pupitress \
    $(workdir)/Notes

%.pdf %.mid: %.ly
    $(LILY_CMD) $<      # cette ligne commence par une tabulation
    if exist "$*.pdf" move /Y "$*.pdf" PDF/ # tabulation au début
    if exist "$*.mid" move /Y "$*.mid" MIDI/ # tabulation au début

notes = \
    cello.ily \
    figures.ily \
    cor.ily \
    hautbois.ily \
    trioCordes.ily \
    alto.ily \
    violonUn.ily \
    violonDeux.ily

$(piece)I.pdf: $(piece)I.ly $(notes)
$(piece)II.pdf: $(piece)II.ly $(notes)
$(piece)III.pdf: $(piece)III.ly $(notes)
$(piece)IV.pdf: $(piece)IV.ly $(notes)

$(piece).pdf: $(piece).ly $(notes)

$(piece)-cello.pdf: $(piece)-cello.ly cello.ily
$(piece)-cor.pdf: $(piece)-cor.ly cor.ily
$(piece)-hautbois.pdf: $(piece)-hautbois.ly hautbois.ily
```

```

$(piece)-alto.pdf: $(piece)-alto.ly alto.ily
$(piece)-violonUn.pdf: $(piece)-violonUn.ly violonUn.ily
$(piece)-violonDeux.pdf: $(piece)-violonDeux.ly violonDeux.ily

.PHONY: score
score: $(piece).pdf

.PHONY: parties
parties: $(piece)-cello.pdf \
         $(piece)-violonUn.pdf \
         $(piece)-violonDeux.pdf \
         $(piece)-alto.pdf \
         $(piece)-hautbois.pdf \
         $(piece)-cor.pdf

.PHONY: mouvements
mouvements: $(piece)I.pdf \
            $(piece)II.pdf \
            $(piece)III.pdf \
            $(piece)IV.pdf

all: score parties mouvements

```

Le *Makefile* suivant convient pour un document `lilypond-book` réalisé avec \LaTeX . Ce projet contiendra un index, ce qui nécessitera de lancer une deuxième fois `latex` pour mettre à jour les liens. Les fichiers résultants iront dans le répertoire `out` pour ce qui est des `.pdf` et dans le répertoire `htmlout` pour ce qui est du `html`.

```

SHELL=/bin/sh
FILE=monprojet
OUTDIR=out
WEBDIR=htmlout
VIEWER=acroread
BROWSER=firefox
LILYBOOK_PDF=lilypond-book --output=$(OUTDIR) --pdf $(FILE).lytex
LILYBOOK_HTML=lilypond-book --output=$(WEBDIR) $(FILE).lytex
PDF=cd $(OUTDIR) && pdflatex $(FILE)
HTML=cd $(WEBDIR) && latex2html $(FILE)
INDEX=cd $(OUTDIR) && makeindex $(FILE)
PREVIEW=$(VIEWER) $(OUTDIR)/$(FILE).pdf &

all: pdf web keep

pdf:
    $(LILYBOOK_PDF) # tabulation en début de ligne
    $(PDF)          # tabulation en début de ligne
    $(INDEX)        # tabulation en début de ligne
    $(PDF)          # tabulation en début de ligne
    $(PREVIEW)      # tabulation en début de ligne

web:
    $(LILYBOOK_HTML) # tabulation en début de ligne
    $(HTML)          # tabulation en début de ligne

```

```

cp -R $(WEBDIR)/$(FILE)/ ./ # tabulation en début de ligne
$(BROWSER) $(FILE)/$(FILE).html & # tabulation en début de ligne

keep: pdf
cp $(OUTDIR)/$(FILE).pdf $(FILE).pdf # tabulation en début de ligne

clean:
rm -rf $(OUTDIR) # tabulation en début de ligne

web-clean:
rm -rf $(WEBDIR) # tabulation en début de ligne

archive:
tar -cvvf monprojet.tar \ # tabulation en début de ligne
--exclude=out/* \
--exclude=htmlout/* \
--exclude=monprojet/* \
--exclude=*midi \
--exclude=*pdf \
--exclude=*~ \
../MonProjet/*

```

AVENIR: faire que ça marche sous Windows

Ce *makefile* n'est malheureusement pas opérationnel sous Windows. La seule alternative qui s'offre aux utilisateurs de Windows consiste à créer un fichier de traitement par lot (*.bat*) qui contienne les différentes commandes successives. Bien que cette manière de procéder ne tienne aucun compte des dépendances entre fichiers, elle permet de réduire le nombre de processus à lancer dans une seule commande. Vous devrez enregistrer les lignes suivantes dans un fichier *construire.bat* ou *construire.cmd*. Ce fichier pourra être exécuté soit en ligne de commande, soit par un double clic sur son icône.

```

lilypond-book --output=out --pdf monprojet.lytex
cd out
pdflatex monprojet
makeindex monprojet
pdflatex monprojet
cd ..
copy out\monprojet.pdf MonProjet.pdf

```

Voir aussi

Manuel d'utilisation : [Section 1.2 \[Utilisation en ligne de commande\]](#), page 1, [Chapitre 3 \[LilyPond-book\]](#), page 8

Annexe A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Annexe B Index de LilyPond

A

ABC..... 13

C

Coda Technology..... 14

coloration syntaxique..... 10

convert-ly..... 4

E

éditeurs..... 10

emacs..... 10

enigma..... 14

ETF..... 14

F

fichier de sortie, taille..... 9

Finale..... 14

M

make..... 18

makefiles..... 18

Manuels..... 1

MIDI..... 11

mise à jour d'anciens fichiers..... 4

mise à jour de fichiers LilyPond..... 4

modes, éditeur..... 10

MusicXML..... 12

O

OpenOffice.org..... 14

P

pointer-cliquer..... 9

programmes externes générant des fichiers LilyPond

..... 14

S

syntaxe, coloration..... 10

T

taille du fichier de sortie..... 9

V

vim..... 10