

Conversion d'un fichier NWC en fichier LY

2 novembre 2021

Les programmes `nwc2ly2.jar` et `nwc2ly3.jar` convertissent les fichiers NWC obtenus à partir du logiciel d'écriture de compositions musicales NoteWorthyComposer en fichiers LilyPond.

Je rappelle que LilyPond qu'on trouve à l'adresse <http://lilypond.org/> est un logiciel de gravure musicale libre, destiné à produire des partitions de qualité optimale.

NoteWorthyComposer qu'on trouve à l'adresse <https://noteworthycomposer.com/> est un shareware permettant d'écrire facilement de la musique. Ce shareware date de 1994, il est simple d'emploi, peu coûteux mais malgré les améliorations successives apportées, reste très moyen au niveau de l'impression graphique obtenue. Il nécessite en plus une intervention lourde pour éviter des collisions de symboles lorsqu'on désire mettre plusieurs voix sur une même portée.

1 Comment convertir un fichier

Le programme de conversion fonctionne avec des fichiers NWCTXT obtenus avec NoteWorthyComposer version 2.7.

1. Pour obtenir un tel fichier il suffit, dans le programme NoteWorthyComposer d'appeler la commande `save as...` du menu `File` et de choisir le type `NWC Text File`.
Si vous utilisez une version plus ancienne le programme de conversion ne fonctionnera pas car il analyse la première ligne du fichier qui devra commencer obligatoirement par :

`!NoteWorthyComposer(2.7`

Si ce n'est pas le cas, on peut toujours « tricher » c'est à dire modifier la première ligne et encoder le fichier en utf8, sous Windows Notepad++ fait ça très bien.

2. On clique ensuite sur le programme `nwc2ly2.jar` ou `nwc2ly3.jar` pour le lancer. On choisit le fichier d'extension `.nwctxt` que l'on veut convertir et le programme génère automatiquement un fichier avec l'extension `.ly`.
3. Je conseille ensuite d'utiliser un « intégrateur », par exemple Frescobaldi pour traiter et compiler le fichier `.ly` obtenu.

Remarques :

- On peut aussi lancer `nwc2ly2.jar` en mode commande, syntaxe :
 - `java -jar nwc2ly2.jar`
 - `java -jar nwc2ly2.jar <fichier>`

- `java -jar nwc2ly2.jar <fichier nwctxt> <fichier ly>`
- `nwc2ly2.jar` ne teste pas la cohérence des données. Il est possible qu'il y ait par exemple un mauvais enchaînement des voix musicales ou encore des barres de reprises mal positionnées. Ceci peut conduire à la création d'un fichier `.ly` qui se compile mal, voir à une erreur de conversion.

2 Structure des fichiers

Le fichier source NWCTXT est un fichier texte. S'il possède des paroles accentuées c'est le codage utf8 qui est utilisé.

Dans ce fichier on observe d'abord une entête contenant, en autres, le titre, l'auteur et le compositeur. Suivent alors une ou plusieurs voix avec aussi des paramètres généraux. Dans une voix donnée un paramètre général est important c'est son articulation avec la voix suivante. Cela permet de distinguer les accompagnements piano, les chœurs SATB et l'écriture de plusieurs voix sur une même portée.

`nwc2ly2.jar` et `nwc2ly3.jar` restituent à peu près cette structure. Ils définissent une entête `\header{...}` puis des voix suivies éventuellement de paroles. Dans un troisième temps ils précisent l'articulation de toutes ces voix dans la ligne `\score{...}`. C'est là qu'on peut agir si on veut modifier la structure générale de la partition.

Remarque : Si le nom de la voix commence par une lettre minuscule ou s'il contient des chiffres alors le convertisseur ajoute des guillemets pour éviter les erreurs de compilation.

Par exemple le fichier `abendlied-extrait.ly` aura la structure suivante :

```
\version "2.18.2"
\paper{
  #(set-paper-size "a4" 'portrait)
  top-margin = 10
  make-footer=##f
  %ragged-last-bottom = ##f
}

\header{
  title = "Abendlied"
  composer = ""
  poet = "Joseph Rheinberger (1839 - 1901)"
  copyright = ""
}

caesura = ^\markup {\musicglyph #"scripts.caesura.straight"}

"S1" = {...}
\addlyrics {...}

"S2" = {...}
\addlyrics {...}

A = {...}
\addlyrics {...}
```

```

"T1" = {...}
\addlyrics {...}

"T2" = {...}
\addlyrics {...}

B = {...}
\addlyrics {...}

Structure = {s4*44 }

\score {<<
\new ChoirStaff <<\new Staff \"S1"
\new Staff \"S2"
\new Staff \A
\new Staff \"T1"
\new Staff \"T2"
\new Staff \B
>> >>}

\layout {#(layout-set-staff-size 20)}

```

et on peut améliorer en retirant les objets en doublon. Ces doublons ne sont utiles que si on neutralise une ou plusieurs voix. Exemple de doublons : `\autoBeamOn`, barre de mesure finale `\bar" | . "`.

Par ailleurs `Structure = {s4*44}` n'a pas d'utilité ici (cf paragraphe sur les reprises) on peut supprimer.

3 Notes et accords, plusieurs voix sur une même portée

Sous `NoteWorthyComposer` les notes sont codées par la position sur la portée, la position 0 correspondant à la ligne du milieu c'est à dire un « si » dans le cas d'une portée en clef de sol. L'inconvénient de ce codage est que la valeur réelle de cette note dépend de la clef utilisée, de l'armure et de la présence d'une note altéré avant.

Sous `lilypond` le codage est absolu. L'apparence de cette note sera déterminée uniquement à l'impression. On peut donc facilement changer la clef sans quasiment rien modifier. Exemple dans `AyLindaAmiga-extrait.ly` on a classiquement en fin de fichier :

```

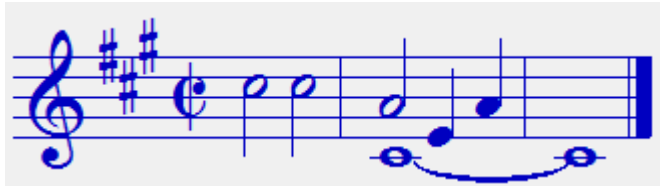
\score {<<
\new Staff \S
\new Staff \A
\new Staff \T
\new Staff \B
>>}

```

Dans `AyLindaAmiga-extrait1.ly` On a regroupé les paroles car elles sont identiques et les voix Soprano et Alto d'une part et Ténor et Basse d'autre part. Les clefs utilisées dans la voix des ténors et des basses sont différentes, cela ne pose aucun problème.

```
\score {<<
\new Staff <<\new Voice\S \new Voice\A >>
\new Staff << \new Voice\T \new Voice\B >>
>>}
```

Remarque : NoteWorthyComposer accepte des accords avec des notes de durées différentes.



Le convertisseur traduit les lignes suivantes :

```
|Note|Dur:Half|Pos:1
|Note|Dur:Half|Pos:1
|Bar
|Chord|Dur:Half|Pos:-1|Opts:Stem=Up|Dur2:Whole|Pos2:-6^
|Note|Dur:4th|Pos:-4|Opts:Stem=Up
|Note|Dur:4th|Pos:-1|Opts:Stem=Up
|Bar
|Note|Dur:Whole|Pos:-6
```

par :

```
cis''2 cis''2
<< {<cis' ~ >1 } {\new Voice <a' >2 e'4 a'4 } >>
cis'1 \bar"|."
```

mais dans le cas de lignes mélodiques distinctes sur toute la partition je conseille d'utiliser des voix séparées.

Voir adest_fideles-extrait.nwctxt pour des séparations locales de voix. Le fichier adest_fideles-extrait.ly peut être amélioré par utilisation de \voiceOne, \voiceTwo et \oneVoice.

Pour mettre deux voix sur une même portée, il suffit, sous NWC, de les créer, de cocher la case Layer with Next Staff dans les propriétés de la première voix. NWC montre alors les deux voix sur une même portée ou sur des portées séparées mais sans gérer les conflits éventuels de notes.

Depuis le 02/11/21, le programme nwc2ly3.jar gère très bien les voix en générant des \voiceOne, \voiceTwo et \oneVoice aux bons endroits à condition de bien définir où seront les silences invisibles et la position des hampes (Up ou Down).

Dans l'exemple ci-dessous (o_createur-mendelssohn.nwctxt) il y a deux voix. Pour que nwc2ly3.jar puisse gérer convenablement les \voiceOne, \voiceTwo et \oneVoice, dans la première voix, on ne précise pas la position des hampes dans les mesures 1 à 3. Par contre on les a mis en position Up dans la mesure 4. Dans la seconde voix, on met toutes les hampes en position Down.



Ceci génère alors le code suivant :

```
...
"d" = {\autoBeamOn \oneVoice \clef treble \key aes \major \time 3/8
r8 <aes c' >8 <bes des' >8
<c' ees' >4 <ees' bes' >8
<c' aes' >4 r8
\voiceOne des''4 des''8
...
"d1" = {\autoBeamOn \oneVoice \clef treble \key aes \major \time 3/8
\voiceTwo
s1*3/8*3 r8 aes'8 g'8
...
\score {<<
\new Staff << \"d" \\\ "d1" \\\ >>
>>}
...
```

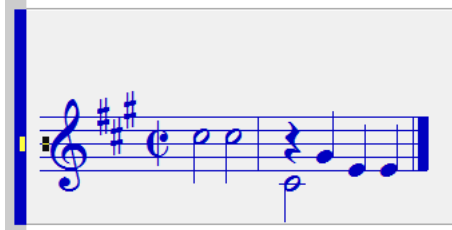
Notez, pour finir, que le sens des hampes dans NWC s'appliquent même aux rondes et silences et qu'il est parfaitement défini dans les regroupements de notes en ligature (commande Beam). Il conviendra donc d'éviter ce type de groupement avant d'utiliser `nwc2ly3.jar`. Notez aussi que, dans l'exemple précédent, on peut retirer le `\oneVoice` dans la ligne `"d1" = {` et remplacer `\new Staff << \"d" \\\ "d1" \\\ >>` par `\new Staff << \"d" \\\ "d1" >>`.

4 Les silences

Le convertisseur remplace les silences par des symboles, par exemple `r4` pour un soupir, `R1` pour une pause, `R1*3/4` si la mesure a trois temps.

Si le silence est invisible, comme par exemple dans `|Rest|Dur:4th|Visibility:Never` alors c'est la lettre "s" et non "r" ou "R" qui sera utilisée. Ainsi la ligne qui précède sera transformée en s4.

Par ailleurs `NoteWorthyComposer` accepte aussi la juxtaposition d'un silence et d'une note ou d'un accord. Ainsi :



qui s'écrit :

```
|Note|Dur:Half|Pos:1
|Note|Dur:Half|Pos:1
|Bar
|RestChord|Dur:4th|Opts:Stem=Up,ArticulationsOnStem|Dur2:Half|Pos2:-6
|Note|Dur:4th|Pos:-2|Opts:Stem=Up
|Note|Dur:4th|Pos:-4
|Note|Dur:4th|Pos:-4
```

sous `NoteWorthyComposer` deviendra :

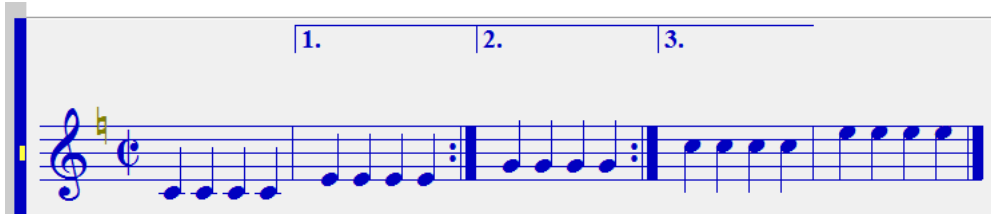
```
cis''2 cis''2
<< {<cis' >2 } {\new Voice r4 gis'4 } >> e'4 e'4 \bar"|."
```

sous `LilyPond`.

5 Les reprises

J'ai fait le choix de ne gérer que les barres de reprise en ignorant les `Special Ending...` de `NWC` et de pas faire de distinctions entre les barres de reprise locales et globales. L'usage n'est pas assez rigoureux et peut conduire à beaucoup d'erreurs. On peut facilement dessiner les barres de reprise qui s'écrivent sous `lilypond: \bar":|.` ou `\bar":|:.`

Néanmoins si on désire gérer les identificateurs `\volta` et `\alternative` et donc éventuellement les `Special Ending...` on peut utiliser la ligne `Structure = {...}` en la regroupant avec une voix déjà définie.



est converti en

```
V = {\autoBeamOn \clef treble \key c \major \time 2/2
c'4 c'4 c'4 c'4
e'4 e'4 e'4 e'4 \bar":|."
```

```
g'4 g'4 g'4 g'4 \bar":|."
c''4 c''4 c''4 c''4
e''4 e''4 e''4 e''4 \bar"|."}
```

```
Structure = {\repeat volta 3 {s4*4} \alternative { {s4*4} {s4*4} {s4*8} } }
```

```
\score {<<
\new Staff \V
>>}
```

On peut alors modifier la partie `\score{...}` de la façon suivante :

```
\score {<<
\new Staff <<\V \Structure>>
>>}
```

Et encore améliorer le résultat en modifiant légèrement `Structure = {...}`.

```
Structure = {\repeat volta 3 {s4*4} \alternative {{s4*4}{s4*4}{s4*4}} s4*4 }
```

6 Les paroles

Elles sont mises au format LilyPond en utf8. Si le couplet démarre par un numéro, 1. _ par exemple alors le convertisseur le transforme en `\set stanza = #"1."`. On peut obtenir aussi cette gestion de numéro en utilisant l'objet `VerseNumbers` du sous-menu `Insert->Object`.

Remarques :

- si on retrouve les mêmes paroles dans chaque voix on peut alors créer la macro `paroles = \lyricmode{...}` puis en-dessous de chaque voix écrire `\addlyrics {\paroles}`.
- si le mode adopté est `\autoBeamOn` alors pour obtenir une syllabe sur plusieurs notes il suffit les relier mais si le mode est `\autoBeamOff` alors la syllabe portera sur plusieurs notes si elles sont reliées par des liaisons ordinaires ou des ligatures. C'est assez standard en musique.

7 Divers

- Le convertisseur reconnaît les `Multi-mesures` `Rest`, les `Performance Style...`, les différents styles de barres de mesure, les « dynamiques » (piano, pianissimo, forte...), les « crescendo » et autres modifications de dynamique, les `Flow Direction` (coda, segno), les `Sustain Pedal`, les indications de tempo et leurs modification (`Tempo Variance`), les `Time Signature`, les `Text...`
- Le convertisseur reconnaît aussi les liaisons (« Tie » et « Slur »), « Stacatto » et autres « Marcato », l'arpeggio, le glissando, le trill et les numéros de couplets (`Insert->Object...`

8 Fragments de Frescobaldi

Il s'agit de petits programmes permettant de faciliter la saisie des partitions sous Frescobaldi.

Pour charger `frescobaldi-fragments.xml`, rendez-vous sur

Fragments->Gérer les fragments->importer.

Je propose 2 «Fragments», un pour créer facilement des trémolos, un autre pour créer un `\partial` en tête des portées.

- Pour obtenir ce type d'écriture :



qu'on ne peut imprimer sous NoteWorthyComposer je propose la procédure suivante :

On écrit directement :



Après conversion on obtient :

```
A = {\autoBeamOn \clef treble \key c \major \time 4/4  
c'4 g'4 c'4 g'4 \bar"|."}
```

Il suffit alors de sélectionner à la souris le premier `c'4 g'4`, de faire Ctrl+&, de rentrer 3 dans la boîte de dialogue et on obtient alors un trémolo de triples croches, la séquence précédente étant alors remplacée automatiquement par `\repeat tremolo 8 { c'32 g'32 }`.

- NoteWorthyComposer ne gère pas non plus les `\partial`. Soit on écrit directement :



mais la structure sera détruite lorsqu'on lancera un audit sur les barres de mesure. Il y aura aussi un décalage au moment de la conversion car la gestion des barres de mesure est automatique sous LilyPond. Aussi je préconise l'ajout de silences invisibles :



Après conversion on obtient :

```
A = {\autoBeamOn \clef treble \key c \major \time 4/4  
s2 s4 c'4  
e'4 g'4 c''4 c''4 \bar"|."}
```

avec donc un espace trop important au début de portée. Je propose donc qu'on sélectionne le contenu de la première mesure à savoir `s2 s4 c'4` et qu'on lance la commande Partial (Ctrl+é). La séquence précédente sera alors convertie en `\partial 4 c'4`. Si on a plusieurs portées il faut évidemment le faire sur toutes et ne pas oublier le `Structure = {`.