# MIDI test suite

crescendo-gap-compatible-target-midi.ly
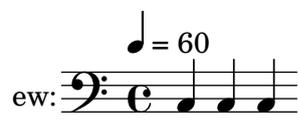
crescendo-return-crescendo-midi.ly
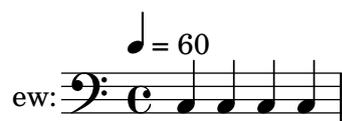
crescendo-return-louder-target-midi.ly
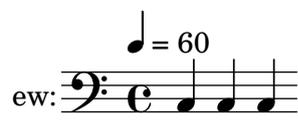

crescendo-return-softer-target-midi.ly
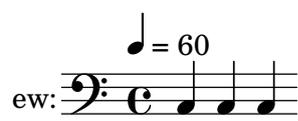
crescendo-return-unspecified-target-midi.ly
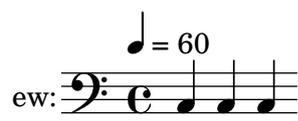
crescendo-single-compatible-target-midi.ly
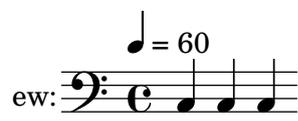

crescendo-single-unspecified-target-midi.ly
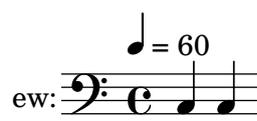
decrescendo-multiple-compatible-target-midi.ly
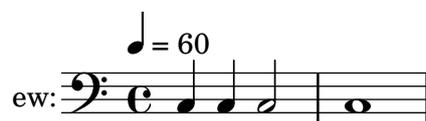
```
decrescendo-single-contrary-target-midi.ly
```

ew: [musical staff: bass clef, ♩ = 60, common time, four notes]

```
dynamic-initial-midi.ly
```

ew: [musical staff: bass clef, ♩ = 60, common time, three notes]
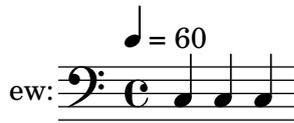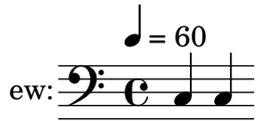
```
dynamic-voices-sequential-midi.ly
```

:A [musical staff: bass clef, ♩ = 60, common time, two notes]
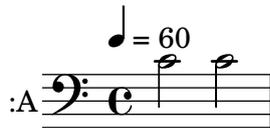
```
dynamic-voices-simultaneous-midi.ly
```

:A [musical staff: treble clef, ♩ = 60, common time, notes]

keys work in MIDI, this is d-minor

```
key-initial-midi.ly
```

trackB: [musical staff: treble clef, one flat, ♩ = 60, common time, notes]
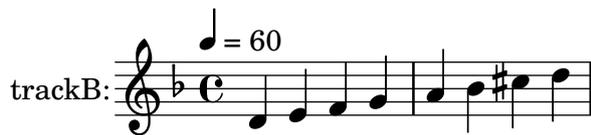
Midi2ly –key works on all staves, this is G major (–key=1)

```
key-option-all-staves-midi.ly
```

treble:one [musical staff: treble clef, one sharp, ♩ = 60, common time]

bass:two [musical staff: bass clef, one sharp, common time]
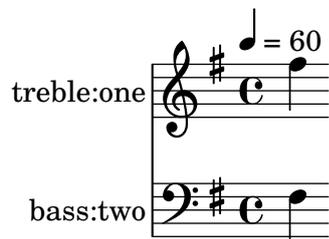
midi2ly's option `--key` works, this is F major.

```
key-option-midi.ly
```

trackB:voiceA [musical staff: treble clef, one flat, ♩ = 60, common time, notes]
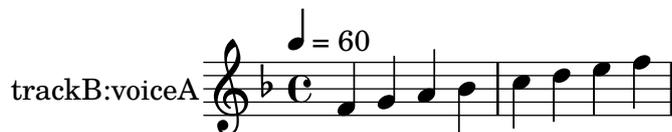
Lyrics are preserved

`lyrics-addlyrics-midi.ly`

:uniqueContext0

Girls and boys come out to play,

`midi-overlapping-notes-midi.ly`

Partcombined music is preserved

`partcombine-midi.ly`

staff:two

midi2ly's option `--duration-quant` preserves first note length (16).

`quantize-duration-2-midi.ly`

trackB:

midi2ly's option `--duration-quant` quantizes durations of notes.

`quantize-duration-midi.ly`

trackB:voiceA

midi2ly's option `--start-quant` quantizes start of notes.

`quantize-start-midi.ly`

trackB:voiceA

LilyPond respects rests, also when there are dynamics

`rest-dynamic-midi.ly`

midi2ly identifies rests

`rest-midi.ly`

trackB:voiceA
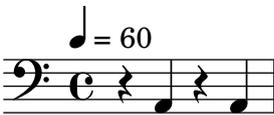
If a score has a `\header` block which defines a title, this title should override any title defined in a `\header` block of the score's enclosing `\bookpart` or `\book` (or a title defined in a top-level `\header` block) when naming the MIDI sequence generated from the score. Otherwise, if the score has no title defined, the MIDI sequence generated from the score should get named using the title defined in the `\header` block of the nearest enclosing `\bookpart`, `\book`, or top-level scope that contains a title definition.

`sequence-name-scoping-midi.ly`

Book
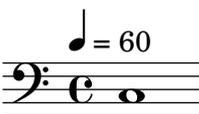
The MIDI sequence generated from a score should get its name from the title defined in the score's `\header` block (if any). The title used for layout can be overridden for MIDI output by specifying a separate `mididitle` in the `\header` block. If the score does not define a title of its own, and has no enclosing `\bookpart`, `\book`, or top-level scope with a `\header` block that defines a title, either, the MIDI sequence should get the default name.

`sequence-name-midi.ly`

Title shared between layout and MIDI
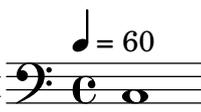
Midi2ly remaps voices correctly to staves in MIDI-files that use instrument`<->`channel mapping when combined with voice`<->`track mapping. TODO: pianostaff

`staff-map-instrument-midi.ly`
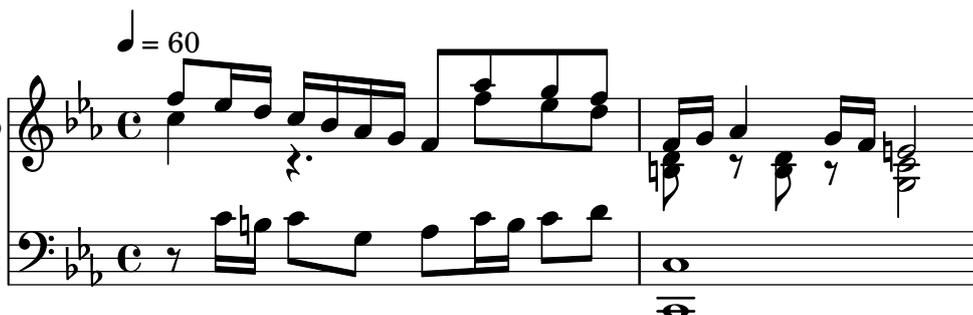
Midi2ly remaps voices correctly to staves in MIDI-files that use voice`<->`channel mapping when combined with staff`<->`track mapping. TODO: pianostaff

`staff-map-voice-midi.ly`

midi2ly maps two voices nicely on one staff as \voiceOne, \voiceTwo

`voice-2-midi.ly`

midi2ly maps four voices nicely on one staff as \voiceOne, \voiceTwo, \voiceThree, \voice-Four

`voice-4-midi.ly`

midi2ly still produces output for a staff with five voices. However, in such cases, most probably the the correct \voiceOne, \voiceX... mapping is lost.

`voice-5-midi.ly`