

MIDI test suite

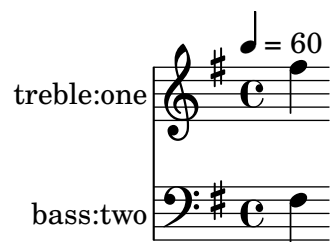
keys work in MIDI, this is d-minor

key-initial-midi.ly



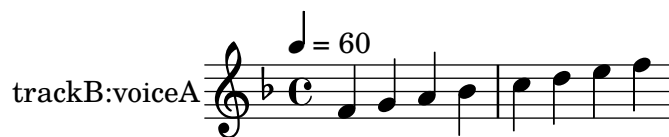
Midi2ly -key works on all staves, this is G major (-key=1)

key-option-all-staves-midi.ly



midi2ly's option --key works, this is F major.

key-option-midi.ly



Lyrics are preserved

lyrics-addlyrics-midi.ly



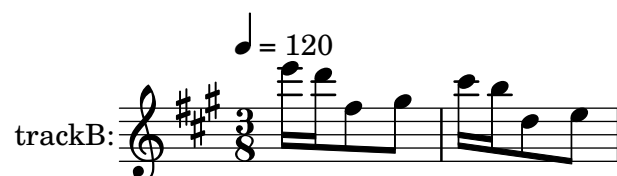
Partcombined music is preserved

partcombine-midi.ly

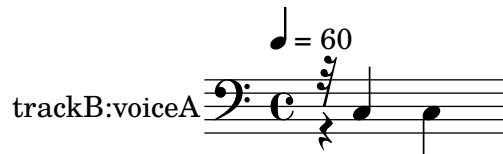


midi2ly's option --duration-quant preserves first note length (16).

quantize-duration-2-midi.ly



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



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



LilyPond respects rests, also when there are dynamics
`rest-dynamic-midi.ly`



midi2ly identifies rests
`rest-midi.ly`



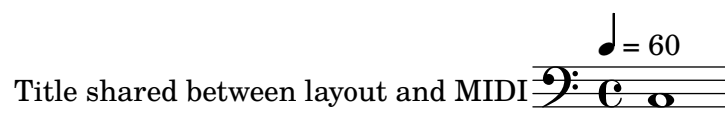
If a score has a `\header` block which defines a title, this title should override any title defined in a `\header` block of the score's enclosing `\bookpart` or `\book` (or a title defined in a top-level `\header` block) when naming the MIDI sequence generated from the score. Otherwise, if the score 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`



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 `midititle` 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`



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, \voiceFour

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

trackB:voiceB

$\text{♩} = 60$

The image shows a musical score for a track labeled 'trackB:voiceB'. The score is written on a single staff with a treble clef and a common time signature (C). Above the staff, there is a tempo marking '♩ = 60'. The music consists of a series of notes: a quarter note on G4, followed by a dotted quarter note on A4, then an eighth note on B4, another eighth note on A4, and a quarter note on G4. The staff ends with a double bar line. Below the staff, there are several musical symbols, including a bass clef, a series of horizontal lines, and a note on a low staff, which appear to be part of a multi-measure rest or a continuation of the piece.