

Bibliothèque MIDICSV

Installation

sudo apt-get install midicsv

Utilisation

midicsv file.midi output.csv

RECORD STRUCTURE

Each record in the CSV representation of a MIDI contains at least three fields:

Track

Numeric field identifying the track to which this record belongs. Tracks of MIDI data are numbered starting at 1. Track 0 is reserved for file header, information, and end of file records.

Time

Absolute time, in terms of MIDI clocks, at which this event occurs. Meta-events for which time is not meaningful (for example, song title, copyright information, etc.) have an absolute time of 0.

Type

Name identifying the type of the record. Record types are text consisting of upper and lower case letters and the underscore (“_”), contain no embedded spaces, and are not enclosed in quotes. **csvmidi** ignores upper/lower case in the **Type** field; the specifications “**Note_on_c**”, “**Note_On_C**”, and “**NOTE_ON_C**” are considered identical.

Records in the CSV file are sorted first by the track number, then by time. Out of order records will be discarded with an error message from **csvmidi**. Following the three required fields are parameter fields which depend upon the **Type**; some **Types** take no parameters. Each **Type** and its parameter fields is discussed below.

Any line with an initial nonblank character of “#” or “;” is ignored; either delimiter may be used to introduce comments in a CSV file. Only full-line comments are permitted; you cannot use these delimiters to terminate scanning of a regular data record. Completely blank lines are ignored.

File Structure Records

0, 0, Header, *format*, *nTracks*, *division*

The first record of a CSV MIDI file is always the **Header** record. Parameters are *format*: the MIDI file type (0, 1, or 2), *nTracks*: the number of tracks in the file, and *division*: the number of clock pulses per quarter note. The **Track** and **Time** fields are always zero.

0, 0, End_of_file

The last record in a CSV MIDI file is always an **End_of_file** record. Its **Track** and **Time** fields are always zero.

Track, 0, Start_track

A **Start_track** record marks the start of a new track, with the *Track* field giving the track number. All records between the **Start_track** record and the matching **End_track** will have the same *Track* field.

Track, Time, End_track

An **End_track** marks the end of events for the specified *Track*. The *Time* field gives the total duration of the track, which will be identical to the *Time* in the last event before the **End_track**.

File Meta-Events

The following events occur within MIDI tracks and specify various kinds of information and actions. They may appear at any time within the track. Those which provide general information for which time is not relevant usually appear at the start of the track with **Time** zero, but this is not a requirement.

Many of these meta-events include a text string argument. Text strings are output in CSV records enclosed in ASCII double quote (") characters. Quote characters embedded within strings are represented by two consecutive quotes. Non-graphic characters in the ISO 8859-1 Latin-1 set are output as a backslash followed by their three digit octal character code. Two consecutive backslashes denote a literal backslash in the string. Strings in MIDI files can be extremely long, theoretically as many as $2^{28}-1$ characters; programs which process MIDI CSV files should take care to avoid buffer overflows or truncation resulting from lines containing long string items. All meta-events which take a text argument are identified by a suffix of "**_t**".

*Track, Time, **Title_t**, Text*

The *Text* specifies the title of the track or sequence. The first **Title** meta-event in a type 0 MIDI file, or in the first track of a type 1 file gives the name of the work. Subsequent **Title** meta-events in other tracks give the names of those tracks.

*Track, Time, **Copyright_t**, Text*

The *Text* specifies copyright information for the sequence. This is usually placed at time 0 of the first track in the sequence.

*Track, Time, **Instrument_name_t**, Text*

The *Text* names the instrument intended to play the contents of this track, This is usually placed at time 0 of the track. Note that this meta-event is simply a description; MIDI synthesisers are not required (and rarely if ever) respond to it. This meta-event is particularly useful in sequences prepared for synthesisers which do not conform to the General MIDI patch set, as it documents the intended instrument for the track when the sequence is used on a synthesiser with a different patch set.

*Track, Time, **Marker_t**, Text*

The *Text* marks a point in the sequence which occurs at the given *Time*, for example "Third Movement".

*Track, Time, **Cue_point_t**, Text*

The *Text* identifies synchronisation point which occurs at the specified *Time*, for example, "Door slams".

*Track, Time, **Lyric_t**, Text*

The *Text* gives a lyric intended to be sung at the given *Time*. Lyrics are often broken down into separate syllables to time-align them more precisely with the sequence.

*Track, Time, **Text_t**, Text*

This meta-event supplies an arbitrary *Text* string tagged to the *Track* and *Time*. It can be used for textual information which doesn't fall into one of the more specific categories given above.

*Track, 0, **Sequence_number**, Number*

This meta-event specifies a sequence *Number* between 0 and 65535, used to arrange multiple tracks in a type 2 MIDI file, or to identify the sequence in which a collection of type 0 or 1 MIDI files should be played. The **Sequence_number** meta-event should occur at **Time** zero, at the start of the track.

*Track, Time, **MIDI_port**, Number*

This meta-event specifies that subsequent events in the **Track** should be sent to MIDI port (bus) *Number*, between 0 and 255. This meta-event usually appears at the start of a track with **Time** zero, but may appear within a track should the need arise to change the port while the track is being played.

*Track, Time, **Channel_prefix**, Number*

This meta-event specifies the MIDI channel that subsequent meta-events and **System_exclusive** events pertain to. The channel *Number* specifies a MIDI channel from 0 to 15. In fact, the *Number* may be as large as 255, but the consequences of specifying a channel number greater than 15 are undefined.

Track, Time, Time_signature, Num, Denom, Click, NotesQ

The time signature, metronome click rate, and number of 32nd notes per MIDI quarter note (24 MIDI clock times) are given by the numeric arguments. *Num* gives the numerator of the time signature as specified on sheet music. *Denom* specifies the denominator as a negative power of two, for example 2 for a quarter note, 3 for an eighth note, etc. *Click* gives the number of MIDI clocks per metronome click, and *NotesQ* the number of 32nd notes in the nominal MIDI quarter note time of 24 clocks (8 for the default MIDI quarter note definition).

Track, Time, Key_signature, Key, Major/Minor

The key signature is specified by the numeric *Key* value, which is 0 for the key of C, a positive value for each sharp above C, or a negative value for each flat below C, thus in the inclusive range -7 to 7. The *Major/Minor* field is a quoted string which will be **major** for a major key and **minor** for a minor key.

Track, Time, Tempo, Number

The tempo is specified as the *Number* of microseconds per quarter note, between 1 and 16777215. A value of 500000 corresponds to 120 quarter notes (“beats”) per minute. To convert beats per minute to a **Tempo value**, take the quotient from dividing 60,000,000 by the beats per minute.

Track, 0, SMPTE_offset, Hour, Minute, Second, Frame, FracFrame

This meta-event, which must occur with a zero **Time** at the start of a track, specifies the SMPTE time code at which it should start playing. The *FracFrame* field gives the fractional frame time (0 to 99).

Track, Time, Sequencer_specific, Length, Data, ...

The **Sequencer_specific** meta-event is used to store vendor-proprietary data in a MIDI file. The *Length* can be any value between 0 and $2^{28}-1$, specifying the number of *Data* bytes (between 0 and 255) which follow. **Sequencer_specific** records may be very long; programs which process MIDI CSV files should be careful to protect against buffer overflows and truncation of these records.

Track, Time, Unknown_meta_event, Type, Length, Data, ...

If **midicsv** encounters a meta-event with a code not defined by the standard MIDI file specification, it outputs an unknown meta-event record in which *Type* gives the numeric meta-event type code, *Length* the number of data bytes in the meta-event, which can be any value between 0 and $2^{28}-1$, followed by the *Data* bytes. Since meta-events include their own length, it is possible to parse them even if their type and meaning are unknown. **csvmidi** will reconstruct unknown meta-events with the same type code and content as in the original MIDI file.

Channel Events

These events are the “meat and potatoes” of MIDI files: the actual notes and modifiers that command the instruments to play the music. Each has a MIDI channel number as its first argument, followed by event-specific parameters. To permit programs which process CSV files to easily distinguish them from meta-events, names of channel events all have a suffix of “_c”.

Track, Time, Note_on_c, Channel, Note, Velocity

Send a command to play the specified *Note* (Middle C is defined as *Note* number 60; all other notes are relative in the MIDI specification, but most instruments conform to the well-tempered scale) on the given *Channel* with *Velocity* (0 to 127). A **Note_on_c** event with *Velocity* zero is equivalent to a **Note_off_c**.

https://www.inspiredacoustics.com/en/MIDI_note_numbers_and_center_frequencies

-> Midi note number to note name

Track, Time, Note_off_c, Channel, Note, Velocity

Stop playing the specified *Note* on the given *Channel*. The *Velocity* should be zero, but you never know what you'll find in a MIDI file.

Track, Time, Pitch_bend_c, Channel, Value

Send a pitch bend command of the specified *Value* to the given *Channel*. The pitch bend *Value* is a 14 bit unsigned integer and hence must be in the inclusive range from 0 to 16383. The value 8192 indicates no pitch bend; 0 the lowest pitch bend, and 16383 the highest. The actual change in pitch these values produce is unspecified.

Track, Time, Control_c, Channel, Control_num, Value

Set the controller *Control_num* on the given *Channel* to the specified *Value*. *Control_num* and *Value* must be in the inclusive range 0 to 127. The assignment of *Control_num* values to effects differs from instrument to instrument. The General MIDI specification defines the meaning of controllers 1 (modulation), 7 (volume), 10 (pan), 11 (expression), and 64 (sustain), but not all instruments and patches respond to these controllers. Instruments which support those capabilities usually assign reverberation to controller 91 and chorus to controller 93.

Track, Time, Program_c, Channel, Program_num

Switch the specified *Channel* to program (patch) *Program_num*, which must be between 0 and 127. The program or patch selects which instrument and associated settings that channel will emulate. The General MIDI specification provides a standard set of instruments, but synthesisers are free to implement other sets of instruments and many permit the user to create custom patches and assign them to program numbers.

Apparently, due to instrument manufacturers' skepticism about musicians' ability to cope with the number zero, many instruments number patches from 1 to 128 rather than the 0 to 127 used within MIDI files. When interpreting *Program_num* values, note that they may be one less than the patch numbers given in an instrument's documentation.

Track, Time, Channel_aftertouch_c, Channel, Value

When a key is held down after being pressed, some synthesisers send the pressure, repeatedly if it varies, until the key is released, but do not distinguish pressure on different keys played simultaneously and held down. This is referred to as "monophonic" or "channel" aftertouch (the latter indicating it applies to the *Channel* as a whole, not individual note numbers on that channel). The pressure *Value* (0 to 127) is typically taken to apply to the last note played, but instruments are not guaranteed to behave in this manner.

Track, Time, Poly_aftertouch_c, Channel, Note, Value

Polyphonic synthesisers (those capable of playing multiple notes simultaneously on a single channel), often provide independent aftertouch for each note. This event specifies the aftertouch pressure *Value* (0 to 127) for the specified *Note* on the given *Channel*.

System Exclusive Events

System Exclusive events permit storing vendor-specific information to be transmitted to that vendor's products.

Track, Time, System_exclusive, Length, Data, ...

The *Length* bytes of *Data* (0 to 255) are sent at the specified *Time* to the MIDI channel defined by the most recent **Channel_prefix** event on the *Track*, as a System Exclusive message. Note that *Length* can be any value between 0 and $2^{28}-1$. Programs which process MIDI CSV files should be careful to protect against buffer overflows and truncation of these records.

Track, Time, System_exclusive_packet, Length, Data, ...

The *Length* bytes of *Data* (0 to 255) are sent at the specified *Time* to the MIDI channel defined by the most recent **Channel_prefix** event on the *Track*. The *Data* bytes are simply blasted out to the MIDI bus without any prefix. This message is used by MIDI devices which break up long system exclusive message into small packets, spaced out in time to avoid overdriving their modest microcontrollers.

Note that *Length* can be any value between 0 and $2^{28}-1$. Programs which process MIDI CSV files should be careful to protect against buffer overflows and truncation of these records.

Exemple avec le fichier "alb_esp2.mid" :

```
0, 0, Header, 1, 8, 480
1, 0, Start_track
1, 0, Title_t, "Espana Op. 165"
1, 0, Title_t, "Tango"
1, 0, Copyright_t, "Copyright © 2001 by Bernd Krueger"
1, 0, Text_t, "Isaac Albeniz"
1, 0, Text_t, "Andantino"
1, 0, Text_t, "Fertiggestellt 28.01.2001\012"
1, 0, Text_t, "Normierung: 23.12.2002\012"
1, 0, Text_t, "Update am 29.8.2010\012"
1, 0, Text_t, "Dauer: 2:08 Minuten\012"
1, 0, SMPTE_offset, 96, 0, 3, 0, 0
1, 0, Time_signature, 2, 2, 24, 8
1, 0, Key_signature, 2, "major"
1, 0, Tempo, 958466
1, 360, Tempo, 932836
1, 720, Tempo, 1069138
...
1, 60480, End_track
2, 0, Start_track
2, 0, Title_t, "Piano right"
2, 0, Program_c, 0, 0
2, 0, Control_c, 0, 7, 100
2, 0, Control_c, 0, 10, 64
2, 0, Text_t, "bdca426d104a26ac9dcb070447587523"
2, 0, Control_c, 0, 91, 127
2, 360, Note_on_c, 0, 57, 49
2, 480, Note_on_c, 0, 57, 0
2, 480, Note_on_c, 0, 69, 55
...
2, 59520, End_track
3, 0, Start_track
3, 0, Title_t, "Piano left"
3, 0, Program_c, 0, 0
3, 0, Control_c, 0, 7, 100
3, 0, Control_c, 0, 10, 64
3, 0, Control_c, 0, 64, 127
3, 0, Note_on_c, 0, 38, 53
3, 0, Note_on_c, 0, 50, 53
3, 720, Note_on_c, 0, 50, 0
3, 720, Note_on_c, 0, 38, 0
```

3, 720, Note_on_c, 0, 33, 51
3, 720, Note_on_c, 0, 45, 51
3, 960, Note_on_c, 0, 45, 0
...
3, 60107, End_track
4, 0, Start_track
4, 0, Title_t, "Albeniz :Espana Op. 165 Tango"
4, 0, End_track
5, 0, Start_track
5, 0, Title_t, "Copyright © 2001 by Bernd Krüger"
5, 0, End_track
6, 0, Start_track
6, 0, Title_t, "<http://www.piano-midi.de>"
6, 0, End_track
7, 0, Start_track
7, 0, Title_t, "Edition: 2010-08-29"
7, 0, End_track
8, 0, Start_track
8, 0, Title_t, "Spur 7"
8, 0, End_track
0, 0, End_of_file