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. Strings in MIDI files can be extremely long, theoretically as many as 2^{28}-1 characters. The length of the text is stored as a series of bytes which is called a variable length quantity. Only the first 7 bits of each byte is significant (right-justified; sort of like an ASCII byte). So, if you have a 32-bit value, you have to unpack it into a series of 7-bit bytes (ie, as if you were going to transmit it over midi in a SYSEX message). Of course, you will have a variable number of bytes depending upon your value. To indicate which is the last byte of the series, you leave bit #7 clear. In all of the preceding bytes, you set bit #7. So, if a value is between 0-127, it can be represented as one byte. The largest value allowed is 0FFFFFFF, which translates to 4 bytes variable length.

Track name

FF 03 len 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.

Copyright

FF 02 len text

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

Instrument name

FF 04 len 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.

Lyric

FF 05 len 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. See karoke formats.

Marker

FF 06 len text

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

Cue point

FF 07 len text

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

Text

FF 01 len 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.

Sequence number

FF 00 02 *ss ss*
FF 00 00

This optional event must occur at the beginning of a track (ie, before any non-zero time and before any midi events). It specifies the sequence number. The two data bytes *ss ss*, are that number which corresponds to the MIDI Cue message. In a format 2 MIDI file, this number identifies each "pattern" (ie, track) so that a "song" sequence can use the MIDI Cue message to refer to patterns.

If the ss ss numbers are omitted (ie, the second form shown above), then the track's location in the file is used. (ie, The first track chunk is sequence number 0. The second track is sequence number 1. Etc). In format 0 or 1, which contain only one "pattern" (even though format 1 contains several tracks), this event is placed in only the track. So, a group of format 0 or 1 files with different sequence numbers can comprise a "song collection". There can be only one of these events per track chunk in a Format 2. There can be only one of these events in a Format 0 or 1, and it must be in the first track.

Tempo

FF 51 03 tt tt tt

The tempo is specified as the Number of microseconds per quarter note, between 1 and 16777215. A value of 500000 (07 A1 20) 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.

NOTE: If there are no tempo events in a MIDI file, then the tempo is assumed to be 120 BPM.

In a format 0 file, the tempo changes are scattered throughout the one track. In format 1, the very first track should consist of only the tempo (and time signature) events so that it could be read by some device capable of generating a "tempo map". It is best not to place MIDI events in this track. In format 2, each track should begin with at least one initial tempo (and time signature) event.

SMPTE offset

FF 54 05 *hr mn se fr ff*

This meta event is used to specify the SMPTE starting point offset from the beginning of the track. It is defined in terms of hours, minutes, seconds, frames and sub-frames (always 100 sub-frames per frame, no matter what sub-division is specified in the MIDI header chunk). In a format 1 file, the SMPTE OFFSET must be stored with the tempo map (ie, the first track), and has no meaning in any other track.

The *hr* hour byte used to specify the hour offset also specifies the frame rate in the following format: *0rrhhhhh* where *rr* is two bits for the frame rate where 00=24 fps, 01=25 fps, 10=30 fps (drop frame), 11=30 fps and *hhhhh* is five bits for the hour (0-23). The hour byte's top bit is always 0.

The *fr* frame byte's possible range depends on the encoded frame rate in the hour byte. A 25 fps frame rate means that a maximum value of 24 may be set for the frame byte.

The *ff* field contains fractional frames in 100ths of a frame.

Time signature

FF 58 04 *nn dd cc bb*

Time signature is expressed as 4 numbers. *nn* and *dd* represent the "numerator" and "denominator" of the signature as notated on sheet music. The denominator is a negative power of 2: 2 = quarter note, 3 = eighth, etc.

The *cc* expresses the number of MIDI clocks in a metronome click.

The *bb* parameter expresses the number of notated 32nd notes in a MIDI quarter note (24 MIDI clocks). This event allows a program to relate what MIDI thinks of as a quarter, to something entirely different.

For example, 6/8 time with a metronome click every 3 eighth notes and 24 clocks per quarter note would be the following event:

FF 58 04 06 03 18 08

NOTE: If there are no time signature events in a MIDI file, then the time signature is assumed to be 4/4.

In a format 0 file, the time signatures changes are scattered throughout the one track. In format 1, the very first track should consist of only the time signature (and tempo) events so that it could be read by some device capable of generating a "tempo map". It is best not to place MIDI events in this track. In format 2, each track should begin with at least one initial time signature (and tempo) event.

Key signature

FF 59 02 *sf mi*

The key signature is specified by the numeric *sf* 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 *mi* field is a number value which will be 0 for a major key and 1 for a minor key.

Program Name

FF 08 len text

The name of the program (ie, patch) used to play the track. This may be different than the Sequence/Track Name. For example, maybe the name of your sequence (ie, track) is "Butterfly", but since the track is played upon an electric piano patch, you may also include a Program Name of "ELECTRIC PIANO".

Device (port) name

FF 09 len text

The name of the MIDI device (port) where the track is routed. This replaces the "MIDI Port" Meta-Event which some sequencers formally used to route MIDI tracks to various MIDI ports (in order to support more than 16 MIDI channels).

For example, assume that you have a MIDI interface that has 4 MIDI output ports. They are listed as "MIDI Out 1", "MIDI Out 2", "MIDI Out 3", and "MIDI Out 4". If you wished a particular track to use "MIDI Out 1" then you would put a Port Name Meta-event at the beginning of the track, with "MIDI Out 1" as the text.

All MIDI events that occur in the track, after a given Port Name event, will be routed to that port.

In a format 0 MIDI file, it would be permissible to have numerous Port Name events intermixed with MIDI events, so that the one track could address numerous ports. But that would likely make the MIDI file much larger than it need be. The Port Name event is useful primarily in format 1 MIDI files, where each track gets routed to one particular port.

Sequencer specific

FF 7F len 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 (values between 0 and 255) which follow. Sequencer specific records may be very long.

End of Track

FF 2F 00

This event is not optional. It must be the last event in every track. It's used as a definitive marking of the end of a track. Only 1 per track.

The following Meta-Events are considered obsolete and should not be used. Use the Device (port) name Meta-Event instead of the MIDI port Meta-Event.

Channel prefix

FF 20 01 *cc*

The MIDI channel (0-15) contained in this event may be used to associate a MIDI channel with all events which follow, including System exclusive and meta-events. This channel is "effective" until the next normal MIDI event (which contains a channel) or the next MIDI Channel Prefix meta-event. If MIDI channels refer to "tracks", this message may be put into a format 0 file, keeping their non-MIDI data associated with a track.

MIDI port

FF 21 01 *pp*

This optional event which normally occurs at the beginning of a track (ie, before any non-zero time and before any midi events) specifies out of which MIDI Port (ie, buss) the MIDI events in the track go. The data byte *pp*, is the port number, where 0 would be the first MIDI buss in the system.

The MIDI spec has a limit of 16 MIDI channels per MIDI input/output (ie, port, buss, jack, or whatever terminology you use to describe the hardware for a single MIDI input/output). The MIDI channel number for a given event is encoded into the lowest 4 bits of the event's Status byte. Therefore, the channel number is always 0 to 15. Many MIDI interfaces have multiple MIDI input/output busses in order to work around limitations in the MIDI bandwidth (ie, allow the MIDI data to be sent/received more efficiently to/from several external modules), and to give the musician more than 16 MIDI Channels. Also, some sequencers support more than one MIDI interface used for simultaneous input/output. Unfortunately, there is no way to encode more than 16 MIDI channels into a MIDI status byte, so a method was needed to identify events that would be output on, for example, channel 1 of the second MIDI port versus channel 1 of the first MIDI port. This MetaEvent allows a sequencer to identify which track events get sent out of which MIDI port. The MIDI events following a MIDI Port MetaEvent get sent out that specified port.

It is acceptable to have more than one Port event in a given track, if that track needs to output to another port at some point in the track.

The following Meta-Events are not standard but are supported by either MidiKit and QMidi.

M-Live Tag

FF 4B len *tt* text

The text specifies meta tag information for the sequence. This is usually placed at time 0 of the first track in the sequence. The data byte *tt* specifies the tag:

01 = genre
02 = artist
03 = composer
04 = duration (seconds)
05 = bpm (tempo)