MIDI File Format Specifications

Jump to bottom

colxi edited this page on Apr 26, 2018 · 2 revisions

MIDI File Format

Standard MIDI files provide a common file format used by most musical software and hardware devices to store song information including the title, track names, and most importantly what instruments to use and the sequence of musical events, such as notes and instrument control information needed to play back the song. This standardization allows one software package to create and save files that can later be loaded and edited by another completely different program, even on a different type of computer. Almost every software music sequencer is capable of loading and saving standard MIDI files.

File Format Specification

Data Formats

All data values are stored in Big-Endian (most significant byte first) format. Also, many values are stored in a variable-length format which may use one or more bytes per value. Variable-length values use the lower 7 bits of a byte for data and the top bit to signal a following data byte. If the top bit is set to 1, then another value byte follows. Below is a table of examples to help demonstrate how variable length values are used.

Value		**Variable-Length**	
Hex	Bin	Hex	Bin
00	00000000	00	00000000

C8	11001000	8148	10000001 01001000
100000	00010000 00000000 00000000	C08000	11000000 10000000 00000000

Example values and their variable-length equivalents.
A variable-length value may use a maximum of 4 bytes. This means the maximum value that can be represented is 0x0FFFFFFF (represented as 0xFF, 0xFF, 0xFF, 0x7F).

File Structure

MIDI files are organized into data chunks. Each chunk is prefixed with an 8 byte header: 4 byte ID string used to identify the type of chunk followed by a 4 byte size which defines the chunk's length as number of bytes following this chunk's header.

Header Chunk

The header chunk contains information about the entire song including MIDI format type, number of tracks and timing division. There is only one header chunk per standard MIDI file and it always comes first. Before describing each element of the header chunk, here is a chart to help give an overview of the chunk's organization.

Offset	Length	Type	Description	Value
0x00	4	char[4]	chunk ID	"MThd" (0x4D546864)
0x04	4	dword	chunk size	6 (0x00000006)
0x08	2	word	format type	0 - 2
0x10	2	word	number of tracks	1 - 65,535
0x12	2	word	time division	see following text

MIDI Header Chunk Format
Chunk ID and Size
The chunk ID is always "MThd" (0x4D546864) and the size is always 6 because the header chunk always contains the same 3 word values.

Format Type
The first word describes the MIDI format type. It can be a value of 0, 1 or 2 and describes what how the following track information is to be interpreted. A type 0 MIDI file has one track that contains all of the MIDI events for the entire song, including the song title, time signature, tempo and music events. A type 1 MIDI file should have two or more tracks. The first, by convention, contains song information such as the title, time signature, tempo, etc. (more detail in Track Chunk section). The second and following tracks contain a title, musical event data, etc. specific to that track. This closely matches the organization of modern multi-track MIDI sequencers. A type 2 MIDI file is sort of a combination of the other two types. It contains multiple tracks, but each track represents a different sequence which may not necessarily be played simultaneously. This is meant to be used to save drum patterns, or other multi-pattern music sequences.

Number of Tracks
The second word simply defines the number of track chunks that follow this header chunk. A type 0 MIDI file may only contain a value of 1, because they can only contain one track. Type 1 and 2 MIDI files may contain up to 65,536 (0xFFFF) tracks.

Time Division
The third and final word in the MIDI header chunk is a bit more complicated than the first two. It contains the time division used to decode the track event delta times into "real" time. This value is represents either ticks per beat or frames per second. If the top bit of the word (bit mask 0x8000) is 0, the following 15 bits describe the time division in ticks per beat. Otherwise the following 15 bits (bit mask 0x7FFF) describe the time division in frames per second. Ticks per beat translate to the number of clock ticks or track delta positions (described in the Track Chunk section) in every quarter note of music. Common values range from 48 to 960, although newer sequencers go far beyond this range to ease working with MIDI and digital audio together. Frames per second is defined by breaking the remaining 15 bytes into two values. The top 7 bits (bit mask 0x7F00) define a value for the number of SMPTE frames and can be 24, 25, 29 (for 29.97 fps) or 30. The remaining byte (bit mask 0x00FF) defines how many clock ticks or track delta positions there are per frame. So a time division example of 0x9978 could be broken down into it's three parts: the top bit is one, so it is in SMPTE frames per second format, the following 7 bits have a value of 25 (0x19) and the bottom byte has a value of 120 (0x78). This means the example plays at 24 frames per second SMPTE time and has 120 ticks per frame.

Track Chunk

Track chunks contain all of the information for an individual track including, track name and music events. Here is an overview of a track chunk's organization.

Offset	Length	Type	Description	Value

0x00	4	char[4]	chunk ID	"MTrk" (0x4D54726B)
0x04	4	dword	chunk size	see following text
0x08	track event data (see following text)			

MIDI Track Chunk Format
Chunk ID and Size
The chunk ID is always "MTrk" (0x4D54726B) and the size varies depending on the number of bytes used for all of the events contained in the track.

Track Event Data
The track event data contains a stream of MIDI events that define information about the sequence and how it is played. The next section describes the different types of events.

MIDI Events

Track events are used to describe all of the musical content of a MIDI file, from tempo changes to sequence and track titles to individual music events. Each event includes a delta time, event type and usually some event type specific data.
Delta-Times
The event delta time is defined by a variable-length value. It determines when an event should be played relative to the track's last event. A delta time of 0 means that it should play simultaneously with the last event. A track's first event delta time defines the amount of time to wait before playing this first event. Events unaffected by time are still preceded by a delta time, but should always use a value of 0 and come first in the stream of track events. Examples of this type of event include track titles and copyright information. The most important thing to remember about delta times is that they are relative values, not absolute times. The actual time they represent is determined by a couple factors. The time division (defined in the MIDI header chunk) and the tempo (defined with a track event). If no tempo is define, 120 beats per minute is assumed.

Types of Events
There are three types of events: MIDI Control Events, System Exclusive Events and Meta Events.

MIDI Channel Events
Musical control information such as playing a note or adjusting a MIDI channel's modulation value are defined by MIDI Channel Events. Each MIDI Channel Event consists of a variable-length delta time (like all track events) and a two or three byte description which determines the MIDI channel it corresponds to, the type of event it is and one or two event type specific values. Below is a table illustrating how MIDI Channel Events are formatted.

Delta Time	Event Type Value	MIDI Channel	Parameter 1	Parameter 2

variable-length	4 bits	4 bits	1 byte	1 byte

MIDI Channel Event Format

MIDI Channel Events are the most common type of track event and usually make up the bulk of a MIDI file. The following table gives an overview of the seven MIDI Channel Events, listing their numeric value and parameters.

Event Type	Value	Parameter 1	Parameter 2
Note Off	0x8	note number	velocity
Note On	0x9	note number	velocity
Note Aftertouch	0xA	note number	aftertouch value
Controller	0xB	controller number	controller value
Program Change	0xC	program number	not used
Channel Aftertouch	0xD	aftertouch value	not used
Pitch Bend	0xE	pitch value (LSB)	pitch value (MSB)

MIDI Channel Events

Although all of the MIDI Channel Events follow the same basic format, each one requires a bit of explanation. Below is a detailed description of each and how it is used.

Note Off Event

The Note Off Event is used to signal when a MIDI key is released. These events have two parameters identical to a Note On event. The note number specifies which of the 128 MIDI keys is being played and the velocity determines how fast/hard the key was released. The note number is normally used to specify which previously pressed key is being released and the velocity is usually ignored, but is sometimes used to adjust the slope of an instrument's release phase.

Note Off	MIDI Channel	Note Number	Velocity
8 (0x8)	0-15	0-127	0-127

Note Off Event Value Ranges

Note On Event

The Note On Event is used to signal when a MIDI key is pressed. This type of event has two parameters. The note number that specifies which of the 128 MIDI keys is being played and the velocity determines how fast/hard the key is pressed. The note number is normally used to specify the instruments musical pitch and the velocity is usually used to specify the instruments playback volume and intensity.

Note On	MIDI Channel	Note Number	Velocity
9 (0x9)	0-15	0-127	0-127

Note On Event Value Ranges

Note Aftertouch Event

The Note Aftertouch Event is used to indicate a pressure change on one of the currently pressed MIDI keys. It has two parameters. The note number of which key's pressure is changing and the aftertouch value which specifies amount of pressure being applied (0 = no pressure, 127 = full pressure). Note Aftertouch is used for extra expression of particular notes, often introducing or increasing some type of modulation during the instrument's sustain phase

Note Aftertouch	MIDI Channel	Note Number	Amount
10 (0xA)	0-15	0-127	0-127

Note Aftertouch Event Value Ranges

Controller Event

The Controller Event signals the change in a MIDI channels state. There are 128 controllers which define different attributes of the channel including volume, pan, modulation, effects, and more. This event type has two parameters. The controller number specifies which control is changing and the controller value defines it's new setting.

Controller	MIDI Channel	Controller Type	Value
11 (0xB)	0-15	0-127	0-127

Controller Event Value Ranges

Below is a list of the defined MIDI controller types.

Value	Controller Type
0 (0x00)	Bank Select
1 (0x01)	Modulation

2 (0x02)	Breath Controller
4 (0x04)	Foot Controller
5 (0x05)	Portamento Time
6 (0x06)	Data Entry (MSB)
7 (0x07)	Main Volume
8 (0x08)	Balance
10 (0x0A)	Pan
11 (0x0B)	Expression Controller
12 (0x0C)	Effect Control 1
13 (0x0D)	Effect Control 2
16-19 (0x10-0x13)	General-Purpose Controllers 1-4
32-63 (0x20-0x3F)	LSB for controllers 0-31
64 (0x40)	Damper pedal (sustain)
65 (0x41)	Portamento
66 (0x42)	Sostenuto
67 (0x43)	Soft Pedal
68 (0x44)	Legato Footswitch
69 (0x45)	Hold 2
70 (0x46)	Sound Controller 1 (default: Timber Variation)
71 (0x47)	Sound Controller 2 (default: Timber/Harmonic Content)
72 (0x48)	Sound Controller 3 (default: Release Time)
73 (0x49)	Sound Controller 4 (default: Attack Time)
74-79 (0x4A-0x4F)	Sound Controller 5-10
80-83 (0x50-0x53)	General-Purpose Controllers 5-8
84 (0x54)	Portamento Control
91 (0x5B)	Effects 1 Depth (formerly External Effects Depth)

92 (0x5C)	Effects 2 Depth (formerly Tremolo Depth)
93 (0x5D)	Effects 3 Depth (formerly Chorus Depth)
94 (0x5E)	Effects 4 Depth (formerly Celeste Detune)
95 (0x5F)	Effects 5 Depth (formerly Phaser Depth)
96 (0x60)	Data Increment
97 (0x61)	Data Decrement
98 (0x62)	Non-Registered Parameter Number (LSB)
99 (0x63)	Non-Registered Parameter Number (MSB)
100 (0x64)	Registered Parameter Number (LSB)
101 (0x65)	Registered Parameter Number (MSB)
121-127 (0x79-0x7F)	Mode Messages

Defined MIDI Controllers

Program Change Event

The Program Change Event is used to change which program (instrument/patch) should be played on the MIDI channel. This type of event takes only one parameter, the program number of the new instrument/patch.

Program Change	MIDI Channel	Program Number
12 (0xC)	0-15	0-127

Program Change Event Value Ranges

Channel Aftertouch Event

The Channel Aftertouch Event is similar to the Note Aftertouch message, except it effects all keys currently pressed on the specific MIDI channel. This type of event takes only one parameter, the aftertouch amount (0 = no pressure, 127 = full pressure).

Channel Aftertouch	MIDI Channel	Amount
13 (0xD)	0-15	0-127

Channel Aftertouch Event Value Ranges

Pitch Bend Event
The Pitch Bend Event is similar to a controller event, except that it is a unique MIDI Channel Event that has two bytes to describe it's value. The pitch value is defined by both parameters of the MIDI Channel Event by joining them in the format of yyyyyyyxxxxxxx where the y characters represent the last 7 bits of the second parameter and the x characters represent the last 7 bits of the first parameter. The combining of both parameters enables high accuracy values (0 - 16383). The pitch value affects all playing notes on the current channel. Values below 8192 decrease the pitch, while values above 8192 increase the pitch. The pitch range may vary from instrument to instrument, but is usually +/-2 semi-tones.

Pitch Bend	MIDI Channel	Value (LSB)	Value (MSB)
14 (0xE)	0-15	0-127	0-127

Pitch Bend Event Value Ranges

Meta Events
Events that are not to be sent or received over a MIDI port are called Meta Events. These events are defined by an event type value of 0xFF and have a variable size of parameter data which is defined after the event type.

Meta Event	Type	Length	Data
255 (0xFF)	0-255	variable-length	type specific

Meta Event Values
There are currently fifteen defined Meta Events. Each one is described in detail below.

Sequence Number
This meta event defines the pattern number of a Type 2 MIDI file or the number of a sequence in a Type 0 or Type 1 MIDI file. This meta event should always have a delta time of 0 and come before all MIDI Channel Events and non-zero delta time events.

Meta Event	Type	Length	Number (MSB)	Number (LSB)
255 (0xFF)	0 (0x00)	2	0-255	0-255

Sequence Number Meta Event Values
Text Event
This meta event defines some text which can be used for any reason including track notes, comments, etc. The text string is usually ASCII text, but may be any character (0x00-0xFF).

Meta Event	Type	Length	Text
255 (0xFF)	1 (0x01)	string length	ASCII text

Text Meta Event Values
Copyright Notice
This meta event defines copyright information including the copyright symbol © (0xA9), the year and the author. This meta event should always be in the first track chunk, have a delta time of 0 and come before all MIDI Channel Events and non-zero delta time events.

Meta Event	Type	Length	Text
255 (0xFF)	2 (0x02)	string length	ASCII text

Copyright Notice Meta Event Values
Sequence/Track Name
This meta event defines the name of a sequence when in a Type 0 or Type 2 MIDI file or in the first track of a Type 1 MIDI file. It defines a track name when it appears in any track after the first in a Type 1 MIDI file. This meta event should always have a delta time of 0 and come before all MIDI Channel Events and non-zero delta time events.

Meta Event	Type	Length	Text
255 (0xFF)	3 (0x03)	string length	ASCII text

Sequence/Track Name Meta Event Values
Instrument Name
This meta event defines the name of an instrument being used in the current track chunk. This event can be used with the MIDI Channel Prefix meta event to define which instrument is being used on a specific channel.

Meta Event	Type	Length	Text
255 (0xFF)	4 (0x04)	string length	ASCII text

Instrument Name Meta Event Values
Lyrics
This meta event defines the lyrics in a song and are usually used to define a syllable or group of works per quarter note. This event can be used as an equivalent of sheet music lyrics or for implementing a karaoke-style system.

Meta Event	Type	Length	Text

255 (0xFF)	5 (0x05)	string length	ASCII text

Lyrics Meta Event Values

Marker

This meta event marks a significant point in time for the sequence. It is usually found in the first track chunk, but may appear in any one. This event can be useful for marking the beginning/end of a new verse or chorus.

Meta Event	Type	Length	Text
255 (0xFF)	6 (0x06)	string length	ASCII text

Marker Meta Event Values

Cue Point

This meta event marks the start of some type of new sound or action. It is usually found in the first track chunk, but may appear in any one. This event is sometimes used by sequencers to mark when playback of a sample or video should begin.

Meta Event	Type	Length	Text
255 (0xFF)	7 (0x07)	string length	ASCII text

Cue Point Meta Event Values

MIDI Channel Prefix

This meta event associates a MIDI channel with following meta events. It's effect is terminated by another MIDI Channel Prefix event or any non- Meta event. It is often used before an Instrument Name Event to specify which channel an instrument name represents.

Meta Event	Type	Length	Channel
255 (0xFF)	32 (0x20)	1	0-15

MIDI Channel Prefix Meta Event Values

End Of Track

This meta event is used to signal the end of a track chunk and must always appear as the last event in every track chunk.

Meta Event	Type	Length
255 (0xFF)	47 (0x2F)	0

End Of Track Meta Event Values

Set Tempo

This meta event sets the sequence tempo in terms of microseconds per quarter-note which is encoded in three bytes. It usually is found in the first track chunk, time-aligned to occur at the same time as a MIDI clock message to promote more accurate synchronization. If no set tempo event is present, 120 beats per minute is assumed. The following formula's can be used to translate the tempo from microseconds per quarter-note to beats per minute and back.

```
MICROSECONDS_PER_MINUTE = 60000000BPM = MICROSECONDS_PER_MINUTE / MPQNMPQN = MICROSECONDS_PER_MINUTE / BPM
```

Meta Event	Type	Length	Microseconds/Quarter-Note
255 (0xFF)	81 (0x51)	3	0-8355711

Set Tempo Meta Event Values

SMPTE Offset

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). The byte used to specify the hour offset also specifies the frame rate in the following format: 0rrhhhhhh 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 hhhhhh is six bits for the hour (0-23). The hour byte's top bit is always 0. The 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.

Meta Event	Type	Length	Hour	Min	Sec	Fr	SubFr
255 (0xFF)	84 (0x54)	5	0-23 *	0-59	0-59	0-30 *	0-99

SMPTE Offset Meta Event Values, * read preceding text for details

Time Signature
This meta event is used to set a sequences time signature. The time signature defined with 4 bytes, a numerator, a denominator, a metronome pulse and number of 32nd notes per MIDI quarter-note. The numerator is specified as a literal value, but the denominator is specified as (get ready) the value to which the power of 2 must be raised to equal the number of subdivisions per whole note. For example, a value of 0 means a whole note because 2 to the power of 0 is 1 (whole note), a value of 1 means a half-note because 2 to the power of 1 is 2 (half-note), and so on. The metronome pulse specifies how often the metronome should click in terms of the number of clock signals per click, which come at a rate of 24 per quarter-note. For example, a value of 24 would mean to click once every quarter-note (beat) and a value of 48 would mean to click once every half-note (2 beats). And finally, the fourth byte specifies the number of 32nd notes per 24 MIDI clock signals. This value is usually 8 because there are usually 8 32nd notes in a quarter-note. At least one Time Signature Event should appear in the first track chunk (or all track chunks in a Type 2 file) before any non-zero delta time events. If one is not specified 4/4, 24, 8 should be assumed.

Meta Event	Type	Length	Numer	Denom	Metro	32nds
255 (0xFF)	88 (0x58)	4	0-255	0-255	0-255	1-255

Time Signature Meta Event Values
Key Signature
This meta event is used to specify the key (number of sharps or flats) and scale (major or minor) of a sequence. A positive value for the key specifies the number of sharps and a negative value specifies the number of flats. A value of 0 for the scale specifies a major key and a value of 1 specifies a minor key.

Meta Event	Type	Length	Key	Scale
255 (0xFF)	89 (0x59)	2	-7-7	0-1

Key Signature Meta Event Values
Sequencer Specific
This meta event is used to specify information specific to a hardware or software sequencer. The first Data byte (or three bytes if the first byte is 0) specifies the manufacturer's ID and the following bytes contain information specified by the manufacturer. The individual manufacturers may document this information in their respective manuals.

Meta Event	Type	Length	Data
255 (0xFF)	127 (0x7F)	variable-length	any type and amount *

Sequencer Specific Meta Event Values, * read preceding text for details

System Exclusive Events

Also known as SysEx Events, these MIDI events are used to control MIDI hardware or software that require special data bytes that will follow their manufacturer's specifications. Every SysEx event includes an ID that specifies which manufacturer's product is to be the intended receiver. All other products will ignore the event. There are three types of SysEx messages which are used to send data in a single event, across multiple events or authorize the transmission of specific MIDI messages.

Normal SysEx Events

These are the most common type of SysEx event and are used to hold a single block of manufacturer specific data. The first byte is always 0xF0 and the second is a variable-length value that specifies the length of the following SysEx data in bytes. The SysEx data bytes must always end with a 0xF7 byte to signal the end of the message.

SysEx Event	Length	Data
240 (0xF0)	variable-length	data bytes, 0xF7

Normal SysEx Event Values

Divided SysEx Events

A large amount of SysEx data in a Normal SysEx Event could cause following MIDI Channel Events to be transmitted after the time they should be played. This will cause an unwanted delay in play back of the following events. The second type of SysEx Events solve this problem by allowing a large amount of SysEx data to be divided into smaller blocks, transmitted with a delay between each division to allow the transmission of other MIDI events in order to prevent congesting of the limited MIDI bandwidth. The initial Divided SysEx Event follows the same format as a Normal SysEx Event with the exception that the last data byte is not 0xF7. This indicates the the SysEx data is not finished and will be continued in an upcoming Divided SysEx Event. Any following Divided SysEx Events before the final one use the a similar format as the first, only the start byte is 0xF0 instead of 0xF7 to signal continuation of SysEx data. The final block follows the same format as the continuation blocks, except the last data byte is 0xF7 to signal the completion of the divided SysEx data.

SysEx Event	Length	Data
240 (0xF0)	variable-length	data bytes
247 (0xF7)	variable-length	data bytes
247 (0xF7)	variable-length	data bytes, 0xF7

Divided SysEx Event Values

Authorization SysEx Events

The last type of SysEx Event authorizes and enables the transmission of special messages such as Song Position Pointer, MIDI Time Code and Song Select messages. These SysEx Events use the event type value 0xF7.

SysEx Event	Length	Data
247 (0xF7)	variable-length	data bytes

Authorization SysEx Event Values

Pages 1

Find a Page...

MIDI File Format Specifications

MIDI File Format

File Format Specification

Data Formats

File Structure

Header Chunk

Track Chunk

MIDI Events

Clone this wiki locally

https://github.com/colxi/midi-parser-js.wiki.git