
midisnake Documentation

Release 0

Ennis Massey

Sep 24, 2017

Contents:

1	Events	1
2	Parser	5
3	Structure	7
4	Indices and tables	9
	Python Module Index	11

This documentation covers the MIDI event objects supported by MIDISnake

class NoteOn (*data: int*) → None
MIDI NoteOn event

Notes

Subclasses the *midisnake.structure.Event* metaclass

event_name
str – Name of Event

indicator_byte
int – Byte that indicates the MIDI Event type

note_number
int – MIDI note number, between 0 and 127

note_name
str – Musical note name, as returned by *get_note_name()*

note_velocity
int – Volume, between 0 and 127

channel_number
int – MIDI Channel number

raw_data
int – Initial data from MIDI file

class NoteOff (*data: int*) → None
MIDI NoteOff event

Notes

Subclasses the `midisnake.structure.Event` metaclass

event_name

str – Name of Event

indicator_byte

int – Byte that indicates the MIDI Event type

note_number

int – MIDI note number, between 0 and 127

note_name

str – Musical note name, as returned by `get_note_name()`

note_velocity

int – Volume, between 0 and 127

channel_number

int – MIDI Channel number

raw_data

int – Initial data from MIDI file

class PolyphonicAftertouch (*data: int*) → None
MIDI PolyPhonic Aftertouch

Notes

Subclasses the `midisnake.structure.Event` metaclass

event_name

str – Name of Event

indicator_byte

int – Byte that indicates the MIDI Event type

pressure

int – Polyphonic Pressure, between 0 and 127

channel_number

int – MIDI Channel number

raw_data

int – Initial data from MIDI file

class PitchBend (*data: int*) → None
MIDI Pitch Bend

Notes

Subclasses the `midisnake.structure.Event` metaclass

event_name

str – Name of Event

indicator_byte

int – Byte that indicates the MIDI Event type

channel_number

int – MIDI Channel number

bend_amount

int – Amount of bend to apply

raw_data

int – Initial data from MIDI file

CHAPTER 2

Parser

This page contains the documentation for the modules file parser

class `Parser` (*midi_file: `_io.BufferedReader`*) → None

CHAPTER 3

Structure

This documentation covers the internal representations of MIDI constructs, such as Headers, Events e.t.c.

class Header (*data: typing.Union[_io.FileIO, _io.BufferedReader]*) → None

Represents a MIDI file header

length

int – length in bytes of the file, minus the length of the header

format

int – Type of MIDI file, can be one of 0, 1 or 2. 0 and 1 are single file MIDI songs, 2 is comprised of multiple files

ntrks

int – Number of *Track* ‘s in the file

tpqn

float – Ticks per quarter note

class Event (*data: int*) → None

Metaclass representing a MIDI Event. Subclasses must implement the `__process()` function

event_name

str – Name of event

indicator_byte

int – Byte that indicates the MIDI Event type

raw_data

int – Initial data from MIDI file

__process (*data: int*) → None

Processes the given data. Data is in the form of the remaining bytes in the file

Called internally by `__init__`

Parameters *data* (*int*) – Data given to be processed.

classmethod valid (*data: int*) → bool

Used by the parser to determine if the event is applicable :param data: bytes of event :type data: int

Returns Whether the event matches or not

Return type `bool`

class **Track** (*data: typing.Union[_io.FileIO, _io.BufferedReader]*) → None

Represents a MIDI track

track_number

int – Track index. Must be 0 or more

length

int – Length of the track in bytes

events

List[Event] – List of events present in the track

class **VariableLengthValue** (*file_io: typing.Union[_io.BufferedReader, _io.FileIO]*) → None

Parses and stores a MIDI variable length value

length

int – Length, in bytes, of value

raw_data

bytearray – raw data read from file

value

int – Final parsed value

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

m

`midisnake.events`, [1](#)

Symbols

`_process()` (Event method), 7

B

`bend_amount` (PitchBend attribute), 3

C

`channel_number` (NoteOff attribute), 2

`channel_number` (NoteOn attribute), 1

`channel_number` (PitchBend attribute), 2

`channel_number` (PolyphonicAftertouch attribute), 2

E

Event (class in `midisnake.structure`), 7

`event_name` (Event attribute), 7

`event_name` (NoteOff attribute), 2

`event_name` (NoteOn attribute), 1

`event_name` (PitchBend attribute), 2

`event_name` (PolyphonicAftertouch attribute), 2

`events` (Track attribute), 8

F

`format` (Header attribute), 7

H

Header (class in `midisnake.structure`), 7

I

`indicator_byte` (Event attribute), 7

`indicator_byte` (NoteOff attribute), 2

`indicator_byte` (NoteOn attribute), 1

`indicator_byte` (PitchBend attribute), 2

`indicator_byte` (PolyphonicAftertouch attribute), 2

L

`length` (Header attribute), 7

`length` (Track attribute), 8

`length` (VariableLengthValue attribute), 8

M

`midisnake.events` (module), 1

N

`note_name` (NoteOff attribute), 2

`note_name` (NoteOn attribute), 1

`note_number` (NoteOff attribute), 2

`note_number` (NoteOn attribute), 1

`note_velocity` (NoteOff attribute), 2

`note_velocity` (NoteOn attribute), 1

NoteOff (class in `midisnake.events`), 1

NoteOn (class in `midisnake.events`), 1

`ntrks` (Header attribute), 7

P

Parser (class in `midisnake.parser`), 5

PitchBend (class in `midisnake.events`), 2

PolyphonicAftertouch (class in `midisnake.events`), 2

`pressure` (PolyphonicAftertouch attribute), 2

R

`raw_data` (Event attribute), 7

`raw_data` (NoteOff attribute), 2

`raw_data` (NoteOn attribute), 1

`raw_data` (PitchBend attribute), 3

`raw_data` (PolyphonicAftertouch attribute), 2

`raw_data` (VariableLengthValue attribute), 8

T

`tpqn` (Header attribute), 7

Track (class in `midisnake.structure`), 8

`track_number` (Track attribute), 8

V

`valid()` (`midisnake.structure.Event` class method), 7

`value` (VariableLengthValue attribute), 8

VariableLengthValue (class in `midisnake.structure`), 8