

# Event

The Event class describes the properties of events.

**On this page:**

[Event Class](#), [Event Constructor](#)

## Class Hierarchy

- [AudioFile](#)
- [Element](#)
  - [Bus](#)
  - [Effect](#)
  - [Instance](#)
  - [Layer](#)
    - [Program](#)
  - [MidiModule](#)
  - [ModulationMatrixRow](#)
  - [Slot](#)
  - [Zone](#)
- [Event](#)
- [LoadProgress](#)
- [ParameterDefinition](#)

## Classes

### Event Class

#### Description

The state of an Event object is described by the following fields.

**Available in:** Processor.

#### Fields

<b>.type</b>	The type of event. See <a href="#">Event Types</a> for details.	number
<b>.id</b>	The ID of the event.	number
<b>.note</b>	The note number in the range of 0 to 127.	number
<b>.velocity</b>	The note-on velocity in the range of 0 to 127.	number
<b>.controller</b>	The controller number. See <a href="#">Controller Numbers</a> for a description of the different controllers.	number
<b>.value</b>	The value of a controller, pitch bend, or note expression event. The value range depends on the event type.	number
<b>.bend</b>	The value of a pitch bend event in the range of -1.0 to 1.0.	number
<b>.noteExpressionType</b>	The type of note expression event. See <a href="#">Note Expression Types</a> for details.	number
<b>.ppqPosition</b>	The position of the event in ppq relative to the song start. The host must be in in playback. This value will be 0.0 if the host is not in playback.	number

#### Fields per Event Type

Which of the fields are used depends on the [Event Type](#).

	noteOn	noteOff	controller	noteExpression
.type	✓	✓	✓	✓
.id	✓	✓		✓
.note	✓	✓		
.velocity	✓	✓		
.controller			✓	
.value			✓	✓
.bend			✓	
.noteExpressionType				✓
.ppqPosition	✓	✓	✓	✓

## Example

```

-- print the fields of an Event object
function printEventMembers(event)
    print("Event Type:", event.type)
    print("ID:", event.id)
    print("Note #:", event.note)
    print("Velocity:", event.velocity)
    print("Controller #:", event.controller)
    print("Value:", event.value)
    print("Pitch Bend:", event.bend)
    print("Note Expression Type:", event.noteExpressionType)
    print("PPQ:", event.ppqPosition, "\n")
end

function onNote(event)
    printEventMembers(event)
    postEvent(event)
end

function onRelease(event)
    printEventMembers(event)
    postEvent(event)
end

function onController(event)
    printEventMembers(event)
    postEvent(event)
end

function onNoteExpression(event)
    printEventMembers(event)
    -- postEvent(event), not needed for note expression
end

```

[Jump to Top](#)

## Constructors

# Event Constructor

**Event(type)**

## Description

Constructor to create a new [Event](#) object of the specified type.

**Available in:** Processor.

## Arguments

<b>type</b>	The type of event. See <a href="#">Event Types</a> for details.	enum or number
-------------	---	----------------

## Return Values

Returns a new [Event](#) object of the specified type.

The fields of the [Event](#) object must be set after its creation.

## Example

```
-- create new note-on event
function onNote(event)
  local newEvent = Event(EventType.noteOn)
  newEvent.note = event.note + 12
  newEvent.velocity = event.velocity
  local id = postEvent(newEvent)
  waitForRelease()
  releaseVoice(id)
end
```

[Jump to Top](#)