

ModulationMatrixRow

The ModulationMatrixRow class inherits all properties and methods of the [Element](#) class.

On this page:

[ModulationMatrixRow Class](#), [setSource1](#), [setSource2](#), [getSource1](#), [getSource2](#)

Element

[Element Class](#), [findChildren](#), [getChild](#), [getParameter](#), [getParameterDefinition](#), [getParameterNormalized](#), [hasParameter](#), [removeFromParent](#), [setName](#), [setParameter](#), [setParameterNormalized](#)

Class Hierarchy

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

Classes

ModulationMatrixRow Class

Description

The [Element](#) object of the modulation matrix row can be obtained with [getModulationMatrixRow](#). It has the following fields.

Available in: Controller, Processor.

Fields

.rowNumber	Returns the index of the modulation matrix row.	number
.zone	Returns the Zone object of the zone that the modulation matrix row belongs to.	Zone

Example

```
-- get the element object of the first zone in the program
zone = this.program:findZones(true)[1]
-- get the element object of the first modulation matrix row
modRow = zone:getModulationMatrixRow(1)
-- print the zone name and row number of the modulation matrix row
print(modRow.zone.name)
print(modRow.rowNumber)
```

[Jump to Top](#)

Methods

setSource1

```
setSource1(source, sourceInfo1, sourceInfo2)
```

Description

Function to set the 1st modulation source of a row in the modulation matrix. The row is specified with the [Zone](#) object of the zone and the index of the modulation matrix row.

Available in: Controller.

Arguments

source	The modulation source. It can be determined via names or indices. See Modulation Source Types for details. Standard modulation sources like the LFOs or the envelopes can be set directly. Special modulation sources like MIDI controllers or MIDI modules can only be set by also specifying sourceInfo1 and sourceInfo2.	enum or number
sourceInfo1	Optional argument to specify the MIDI controller number or the MIDI module, for example. See example for details.	number or MidiModule , optional
sourceInfo2	Optional argument to select the modulation output of a MIDI module, for example. See example for details.	number, optional

Example

```
-- define the modulation sources and infos in an array
modSources = {
  { source = ModulationSource.lfo1, bipolar = 1 },
  { source = ModulationSource.midiControl, bipolar = 0, sourceInfo1 = 1 },
  { source = ModulationSource.quickControl, bipolar = 1, sourceInfo1 = this.program, sourceInfo2 = 1 },
  { source = ModulationSource.modulationModule, bipolar = 0, sourceInfo1 = this, sourceInfo2 = 1 },
  { source = ModulationSource.modulationModule, bipolar = 0, sourceInfo1 = this, sourceInfo2 = 2 },
  { source = ModulationSource.noteExpression, bipolar = 0, sourceInfo1 = 1 }
}

-- define two modulation outputs for the script module
defineModulation("50%", false)
defineModulation("100%", false)

-- calculate the modulation outputs
function calcModulation()
  return 0.5, 1
end

-- get the element object of the first zone
zone = this.program:findZones(true)[1]

-- assign the modulation sources to source 1 in the modulation rows 1 to 6
for i=1, #modSources do
  local modRow = zone:getModulationMatrixRow(i)
  modRow:setSource1(modSources[i].source, modSources[i].sourceInfo1, modSources[i].sourceInfo2)
  modRow:setParameter("Source1.Polarity", modSources[i].bipolar) -- set the default polarity of the
source
end

-- assign the sample & hold to source 2 in modulation row 1
modRow = zone:getModulationMatrixRow(1)
modRow:setSource2(ModulationSource.sampleAndHold, 0)
```

setSource2

```
setSource2(source, sourceInfo1, sourceInfo2)
```

Description

Function to set the 2nd modulation source of a row in the modulation matrix. The row is specified with the [Zone](#) object of the zone and the index of the modulation matrix row.

Available in: Controller.

Arguments

source	The modulation source. It can be determined via names or indices. See Modulation Source Types for details. Standard modulation sources like the LFOs or the envelopes can be set directly. Special modulation sources like MIDI controllers or MIDI modules can only be set by also specifying sourceInfo1 and sourceInfo2.	enum or number
sourceInfo1	Optional argument to specify the MIDI controller number or the MIDI module, for example.	number or MidiModule , optional
sourceInfo2	Optional argument to select the modulation output of a MIDI module, for example.	number, optional

Example

```
-- define the modulation sources and infos in an array
modSources = {
  { source = ModulationSource.lfo1, bipolar = 1 },
  { source = ModulationSource.midiControl, bipolar = 0, sourceInfo1 = 1 },
  { source = ModulationSource.quickControl, bipolar = 1, sourceInfo1 = this.program, sourceInfo2 = 1 },
  { source = ModulationSource.modulationModule, bipolar = 0, sourceInfo1 = this, sourceInfo2 = 1 },
  { source = ModulationSource.modulationModule, bipolar = 0, sourceInfo1 = this, sourceInfo2 = 2 },
  { source = ModulationSource.noteExpression, bipolar = 0, sourceInfo1 = 1 }
}

-- define two modulation outputs for the script module
defineModulation("50%", false)
defineModulation("100%", false)

-- calculate the modulation outputs
function calcModulation()
  return 0.5, 1
end

-- get the element object of the first zone
zone = this.program:findZones(true)[1]

-- assign the modulation sources to source 1 in the modulation rows 1 to 6
for i=1, #modSources do
  local modRow = zone:getModulationMatrixRow(i)
  modRow:setSource1(modSources[i].source, modSources[i].sourceInfo1, modSources[i].sourceInfo2)
  modRow:setParameter("Source1.Polarity", modSources[i].bipolar) -- set the default polarity of the
source
end

-- assign the sample & hold to source 2 in modulation row 1
modRow = zone:getModulationMatrixRow(1)
modRow:setSource2(ModulationSource.sampleAndHold, 0)
```

getSource1

getSource1()

Description

Function to retrieve the 1st modulation source of a row in the modulation matrix. The row is specified with the [Zone](#) object of the zone and the index of the modulation matrix row.

Available in: Controller.

Return Values

Returns up to three values, i.e., source, sourceInfo1 and sourceInfo2. The number of return values depends on the modulation source. See [Modulation Source Types](#) for details.

Example

```
-- print modulation sources
function printSource(source)
  if source.source == ModulationSource.midiControl then
    print("MIDI Ctrl, Controller Number: "..source.info1)
  elseif source.source == ModulationSource.quickControl then
    print("Quick Ctrl, Layer: "..source.info1.name..", QC Index: "..source.info2)
  elseif source.source == ModulationSource.modulationModule then
    print("MIDI Module, Module: "..source.info1.name..", Output: "..source.info2)
  elseif source.source == ModulationSource.noteExpression then
    print("Note Expression, Custom NE: "..source.info1)
  elseif source.source == ModulationSource.sampleAndHold then
    print("Sample & Hold, Index: "..source.info1)
  else
    print("Standard Modulation, Index: "..source.source)
  end
end

-- get the element object of the first zone
zone = this.program:findZones(true)[1]

-- run through all 32 modulation rows of the zone and print source1 if assigned
for i=1, 32 do
  local modRow = zone:getModulationMatrixRow(i)
  local source1 = {}
  source1.source, source1.info1, source1.info2 = modRow:getSource1()
  if source1.source ~= ModulationSource.unassigned then
    print("Modulation Row "..i..", Source 1:")
    printSource(source1)
  end
end

end
```

[Jump to Top](#)

getSource2

getSource2()

Description

Function to retrieve the 2nd modulation source of a row in the modulation matrix. The row is specified with the [Zone](#) object of the zone and the index of the modulation matrix row.

Available in: Controller.

Return Values

Returns up to three values, i.e., source, sourceInfo1 and sourceInfo2. The number of return values depends on the modulation source. See [Modulation Source Types](#) for details.

The 2nd modulation source has an additional sample & hold. See [Modulation Source Types](#) for details.

Example

```
-- print modulation sources
function printSource(source)
  if source.source == ModulationSource.midiControl then
    print("MIDI Ctrl, Controller Number: "..source.info1)
  elseif source.source == ModulationSource.quickControl then
    print("Quick Ctrl, Layer: "..source.info1.name..", QC Index: "..source.info2)
  elseif source.source == ModulationSource.modulationModule then
    print("MIDI Module, Module: "..source.info1.name..", Output: "..source.info2)
  elseif source.source == ModulationSource.noteExpression then
    print("Note Expression, Custom NE: "..source.info1)
  elseif source.source == ModulationSource.sampleAndHold then
    print("Sample & Hold, Index: "..source.info1)
  else
    print("Standard Modulation, Index: "..source.source)
  end
end

-- get the element object of the first zone
zone = this.program:findZones(true)[1]

-- run through all 32 modulation rows of the zone and print source2 if assigned
for i=1, 32 do
  local modRow = zone:getModulationMatrixRow(i)
  local source2 = {}
  source2.source, source2.info1, source2.info2 = modRow:getSource2()
  if source2.source ~= ModulationSource.unassigned then
    print("Modulation Row "..i..", Source 2:")
    printSource(source2)
  end
end
end
```

[Jump to Top](#)