

Example

Trigger - Add MIDI trigger

by:	Gregory Haynes, Sebastian Beutel June/August 2018
published:	http://forum.avolites.com/viewtopic.php?f=20&t=5726
description:	add midi triggers
remarks:	interesting to predefine some triggers in setup macros

[add](#), [midi](#), [trigger](#)

functions

- [Triggers.SelectMappingByName](#)
- [Triggers.SetPendingTargetHardware](#)
- [Triggers.AddMidiTrigger](#)
- [Math.ToEnum](#)
- [Triggers.SetPendingTargetAction](#)
- [Triggers.SetPendingTargetHandle](#)

affected properties

- [Triggers.SelectedMappingId](#)
- [Triggers.PendingTriggerTarget](#)

identifiers (other than standard)

- [MenuEventTypes](#)

Code

Long lines wrapped for readability - you can as well delete all line wraps.

[addmiditrigger.xml](#)

```
<?xml version="1.0" encoding="utf-8"?>
<avolites.macros>

  <!-- the following will add a MIDI trigger for the first playback
  swap button
      (Hardware) for note 11 (Note On, After Touch and Note Off) -->

  <macro id="Avolites.Macros.AddMidi1" name="Add Midi Trgger Example
  1">
    <sequence>
      <step>Triggers.SelectMappingByName("Your Mapping Name")</step>
      <step>Triggers.SetPendingTargetHardware(
```

```
"PlaybackSelect",
Math.ToEnum(
    "",
    "Avolites.Menus.MenuEventTypes",
    "OnSelect"),
"Playbacks",
0,
"Swop")
</step>
<step>Triggers.AddMidiTrigger(
    Triggers.SelectedMappingId,
    Triggers.PendingTriggerTarget,
    0,
    Math.ToEnum(
        "Avolites.Titan.Controllers.Console",
        "Avolites.Titan.Controllers.Console.PanelMidiCommands",
        "Note"),
    11,
    0,
    127,
    0,
    false)
</step>
</sequence>
</macro>
```

<!-- Add a MIDI trigger for the first preset fader (Hardware) for Control Change 1: -->

```
<macro id="Avolites.Macros.AddMidi2" name="Add Midi Trigger Example 2">
    <sequence>
        <step>Triggers.SelectMappingByName("Your Mapping Name")</step>
        <step>Triggers.SetPendingTargetHardware(
            "PresetFader",
            Math.ToEnum(
                "",
                "Avolites.Menus.MenuEventTypes",
                "OnValueChanged"),
            "Presets",
            0,
            "")
        </step>
        <step>Triggers.AddMidiTrigger(
            Triggers.SelectedMappingId,
            Triggers.PendingTriggerTarget,
            1,
            Math.ToEnum(
                "Avolites.Titan.Controllers.Console",
                "Avolites.Titan.Controllers.Console.PanelMidiCommands",
                "Continuous"),
```

```

    1,
    0,
    127,
    0,
    false)
  </step>
</sequence>
</macro>

<!-- MIDI control change 17 triggers playback button 1 -->

<macro id="Avolites.Macros.AddMidi3" name="Add Midi Trigger Example
3">
  <sequence>
    <step>Triggers.SelectMappingByName("4")</step>
    <step>Triggers.SetPendingTargetHardware("PlaybackSelect",
OnButtonDown, "Playbacks", 1, "")</step>
    <step>Triggers.AddMidiTrigger(
    Triggers.SelectedMappingId,
    Triggers.PendingTriggerTarget,
    1,
    Math.ToEnum(
      "Avolites.Titan.Controllers.Console",
      "Avolites.Titan.Controllers.Console.PanelMidiCommands",
      "Continuous"),
    17,
    0,
    127,
    0,
    false)
    </step>
  </sequence>
</macro>

<!-- MIDI control change 12 triggers playback item on playbacks 1,
action="flash" -->

<macro id="Avolites.Macros.AddMidi4" name="Add Midi Trigger Example
4">
  <sequence>
    <step>Triggers.SelectMappingByName("4")</step>
<step>Triggers.SetPendingTargetHandle("Location=Playbacks,1,1")</step>
    <step>Triggers.SetPendingTargetAction("Flash")</step>
    <step>Triggers.AddMidiTrigger(
    Triggers.SelectedMappingId,
    Triggers.PendingTriggerTarget,
    1,
    Math.ToEnum(
      "Avolites.Titan.Controllers.Console",
      "Avolites.Titan.Controllers.Console.PanelMidiCommands",
      "Continuous"),

```

```
    12,  
    0,  
    127,  
    0,  
    false)  
  </step>  
</sequence>  
</macro>  
</avolites.macros>
```

Explanation

This explains the functional steps within the sequence. For all the other XML details please refer to [Formats and syntax](#)

Here, various triggers are created, simply by the same commands, with different parameters.

- `Triggers.SelectMappingByName()` defines the trigger mapping where the trigger is to be added
- for trigger type=hardware, the triggered hardware is set with `Triggers.SetPendingTargetHardware()`
- for trigger type=item, the triggered item handle is defined with `Triggers.SetPendingTargetHandle()` and the trigger action is defined with `Triggers.SetPendingTargetAction()`
- the actual trigger is then created with `Triggers.AddMidiTrigger` with all the little details (e.g. MIDI channel etc.) as parameter

How to use it

1. [make this macro available](#)

This is best used to prepare some triggers which you always use, in some setup macros.

From:

<https://www.avosupport.de/wiki/> - **AVOSUPPORT**

Permanent link:

<https://www.avosupport.de/wiki/macros/example/addmidittrigger>

Last update: **2019/04/27 22:20**

