Example

Tap specific BPM rate

by:	Sebastian Beutel
published:	February 2018
description:	taps/sets BPM masters to predefined BPM rates
remarks:	Two possible ways - both have their pros and cons, and are discussed below.

tap, bpm, master, speed

The Titan IDs used in this example have changed in Titan v11 (and were different in v10), see titanId

The former macro for v10 can be downloaded here:

tap85bpm.xml

functions

- Masters.TapTempo
- Math.GetCurrentTimeStamp
- Masters.SetSpeed

Code

tapxbpm.xml

```
<?xml version="1.0" encoding="utf-8"?>
<avolites.macros xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Avolites.Menus.xsd">
  <macro id="Avolites.Macros.TapAllBPM60.v11" name="Tap all BPMs 60
(v11)">
    <sequence>
      <step>Masters.TapTempo(1612, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1616, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1620, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1624, Math.GetCurrentTimeStamp())</step>
      <step pause="1.000">Masters.TapTempo(1612,
Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1616, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1620, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1624, Math.GetCurrentTimeStamp())</step>
    </sequence>
  </macro>

  <macro id="Avolites.Macros.TapAllBPM85.v11" name="Tap all BPMs 85
(v11)">
    <sequence>
```

```
      <step>Masters.TapTempo(1612, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1616, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1620, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1624, Math.GetCurrentTimeStamp())</step>
      <step pause="0.706">Masters.TapTempo(1612,
Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1616, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1620, Math.GetCurrentTimeStamp())</step>
      <step>Masters.TapTempo(1624, Math.GetCurrentTimeStamp())</step>
    </sequence>
  </macro>

  <macro id="Avolites.Macros.AllBPM85.v11" name="All BPMs 85 (v11)">
    <sequence>
      <step>Masters.SetSpeed(1612, 85.000)</step>
      <step>Masters.SetSpeed(1616, 85.000)</step>
      <step>Masters.SetSpeed(1620, 85.000)</step>
      <step>Masters.SetSpeed(1624, 85.000)</step>
    </sequence>
  </macro>

  <macro id="Avolites.Macros.AllBPM60.v11" name="All BPMs 60 (v11)">
    <sequence>
      <step>Masters.SetSpeed(1612, 60.000)</step>
      <step>Masters.SetSpeed(1616, 60.000)</step>
      <step>Masters.SetSpeed(1620, 60.000)</step>
      <step>Masters.SetSpeed(1624, 60.000)</step>
    </sequence>
  </macro>

    <macro id="Avolites.Macros.AllBPM0.v11" name="All BPMs 0 (v11)">
    <sequence>
      <step>Masters.SetSpeed(1612, 0.000)</step>
      <step>Masters.SetSpeed(1616, 0.000)</step>
      <step>Masters.SetSpeed(1620, 0.000)</step>
      <step>Masters.SetSpeed(1624, 0.000)</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

The first two macros are essentially identical, except that their target BPM rate is different. What they do is:

- tap all four BPM masters once
- wait a specific time using Step Pause
- tap all masters the 2nd time

Waiting time must be calculated like this:

```
waiting time = 60 / target BPM rate
```

e.g. target BPM rate = 85 ⇒ waiting time = 60/85 = 0.706

The next macros directly set the designated speed.

Pros and Cons

Masters.TapTempo() also sets the edge of the tap as this does exactly as like really tapping the master. But there might be slight deviations since this involves real time on the console - and if there is heavy calculation load, it might happen that the time is not taken too precisely. More importantly, this macro involves a rather long waiting time - and if you trigger the macro again in this time, the result is a completely wrong BPM rate.

Masters.SetSpeed() hasn't the second tap and no waiting time - it always sets an exact BPM rate at once. But it doesn't set the edge. And, even more importantly, it sets the master value to the target value (in this case, 85.000) - regardless what the master is currently set to show. This is, if the masters is set to have BPM on fader then it is all good. But if instead the Multiplier is on the fader then this is set to the value - most likely you'll end with a very high multiplier.

How to use it

- make this macro available
- tap when you need to set the BPM masters to a specific speed