Example

Chase - Double speed

by:	Gregory Haynes, Feb 2016
published:	http://forum.avolites.com/viewtopic.php?f=20&t=3744#p15848
description:	double current chase's BPM rate
remarks:	The problem with this code is that it requires the chase to be set as the selected playback in the editor (used for the Edit Times menu). This means that the Playback View which could be showing a cue list switches to show the chase instead, to get around this I have stored the previous selected playback and restore that afterwards but this does cause the display to flicker. **In version 10.1, one of the affected variables has changed - you need to use the correct macro for your Titan version.**

double, connect, chase, speed, bpm

functions

- ActionScript.SetProperty
- ActionScript.SetProperty.Float
- Math.Min
- Math.IsEqual
- SimpleMath

affected properties

- Handles.SourceHandle
- Playbacks.Editor.SelectedPlayback
- Playbacks.Editor.Times.ChaseSpeed (until version 10.0)
- Playbacks.Editor.Times.PlaybackSpeed (from version 10.1)

This macro changes the speed which is stored with the chase. If you've set the user setting 'Connected View Sets: Temporary Chase Speed' then when applying this macro, the main chase speed is doubled and then overwrites the temp chase speed.

In version 10.1, one of the affected variables has changed - you need to use the correct macro for your Titan version.

Code

until Titan version 10.0

doublechasespeed.xml

```
<?xml version="1.0" encoding="utf-8"?>
<avolites.macros xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:noNamespaceSchemaLocation="Avolites.Menus.xsd">
 <!-- This macro does not work in versions from 10.1 as the property
Playbacks.Editor.Times.ChaseSpeed has been swopped for
Playbacks.Editor.Times.PlaybackSpeed. -->
  <macro name="Chase Speed Double"
id="Avolites.Macros.ChaseSpeedDouble">
    <description>Double the speed of the currently connected
chase.</description>
    <sequence>
      <step>ActionScript.SetProperty("Handles.SourceHandle",
Playbacks.Editor.SelectedPlayback)</step>
<step>ActionScript.SetProperty("Playbacks.Editor.SelectedPlayback",
Chases.ConnectedHandle)</step>
      <step>Math.Min("Playbacks.Editor.Times.ChaseSpeed",
Playbacks.Editor.Times.ChaseSpeed * 2, 3600.0)</step>
      <step condition="Math.IsEqual(Playbacks.Editor.Times.ChaseSpeed,
0.0)">ActionScript.SetProperty.Float("Playbacks.Editor.Times.ChaseSpeed
", 1.0)</step>
<step>ActionScript.SetProperty("Playbacks.Editor.SelectedPlayback",
Handles.SourceHandle)</step>
    </sequence>
  </macro>
</avolites.macros>
```

from Titan version 10.1

doublechasespeed.xml

```
<?xml version="1.0" encoding="utf-8"?>
<avolites.macros xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Avolites.Menus.xsd">
 <!-- This macro does not work in versions below 10.1 as the property
Playbacks.Editor.Times.PlaybackSpeed formerly was
Playbacks.Editor.Times.PlaybackSpeed. -->
  <macro name="Chase Speed Double"
id="Avolites.Macros.ChaseSpeedDouble">
    <description>Double the speed of the currently connected
chase.</description>
    <sequence>
      <step>ActionScript.SetProperty("Handles.SourceHandle",
Playbacks.Editor.SelectedPlayback)</step>
<step>ActionScript.SetProperty("Playbacks.Editor.SelectedPlayback",
Chases.ConnectedHandle)</step>
      <step>Math.Min("Playbacks.Editor.Times.PlaybackSpeed",
Playbacks.Editor.Times.PlaybackSpeed * 2, 3600.0)</step>
      <step
condition="Math.IsEqual(Playbacks.Editor.Times.PlaybackSpeed,
0.0)">ActionScript.SetProperty.Float("Playbacks.Editor.Times.PlaybackSp
```

```
eed", 1.0)</step>
<step>ActionScript.SetProperty("Playbacks.Editor.SelectedPlayback",
Handles.SourceHandle)</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

In version 10.1, the property `Playbacks.Editor.Times.ChaseSpeed` was swopped for `Playbacks.Editor.Times.PlaybackSpeed`. Apart from that, the mechanics of the macro are unchanged.

- `ActionScript.SetProperty(“Handles.SourceHandle”, Playbacks.Editor.SelectedPlayback)`
  temporarily stores the currently selected playback into 'Handles.SourceHandle' from where it will be retrieved when the macro is finished, see Gregory's explanation above
- `ActionScript.SetProperty(“Playbacks.Editor.SelectedPlayback”, Chases.ConnectedHandle)`
  this selects the currently connected chaser for editing
- `Math.Min(“Playbacks.Editor.Times.ChaseSpeed”, Playbacks.Editor.Times.ChaseSpeed * 2, 3600.0)`
  sets the speed of the chaser to its double, or 3600.0, whatever the smaller value - this way it will never be faster than 3600 BPM
- `<step condition=“Math.IsEqual(Playbacks.Editor.Times.ChaseSpeed, 0.0)”>ActionScript.SetProperty.Float(“Playbacks.Editor.Times.ChaseSpeed” , 1.0)</step>`
  if the speed equals 0.0: sets speed to 1.0
- `ActionScript.SetProperty(“Playbacks.Editor.SelectedPlayback”, Handles.SourceHandle)`
  restores the selected playback from before the macro was run

How to use it

Please choose the correct macro for your Titan version.

- make this macro available
- with a chase running and connected (Speed and Cross Fade on the wheels), triggering this macro doubles the speed of the chase
- watch the user setting 'Connected View Sets'. This macro always affects the main chase speed, not the temporary speed. Hence, if there is a temporary chase speed set with the wheels, applying this macro will return to the main chase speed and double it.

From:
https://avosupport.de/wiki/ - **AVOSUPPORT**

Permanent link:
https://avosupport.de/wiki/macros/example/doublechasespeed

Last update: **2019/04/27 11:26**