

Example

PaletteChaseChanger (V3)

by:	Kim Wida, Feb 2018
published:	http://forum.avolites.com/viewtopic.php?f=20&t=5578
description:	two macros: one changes two palettes in blind mode, the other shifts a number of palettes in pairs.
remarks:	slightly corrected by Sebastian Beutel

This is related to [ColourChaseChanger \(V2\)](#).

[palette](#), [chase](#), [blind](#)

functions

- [ActionScript.SetProperty.Boolean](#)
- [Programmer.SetBlindMode](#)
- [Attribute.Mask.IncludeAll](#)
- [Group.RecallGroupNumeric](#)
- [Palette.ApplyPalette](#)
- [Palette.StoreCurrentPaletteReplace](#)
- [Programmer.Editor.Clear](#)
- [Handles.SetSourceHandleFromHandle](#)
- [Palette.StoreCurrentPaletteReplace](#)
- [ActionScript.SetProperty.Enum](#)
- [Handles.CopyDestination](#)
- [Handles.ClearSelection](#)

affected properties

- [Programmer.BlindActive](#)
- [Palette.CurrentPaletteHandle](#)
- [Handles.SourceHandle](#) (*implicitly used*)
- [Handles.OperationMode](#)

Code

palettechasechanger_v3.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="UpTwoColorChange" name="Up Two Change">
    <!-- This swops palette 1,71 with 1,68 and 1,72 with 1,69. -->
    <sequence>
```

```
<step>ActionScript.SetProperty.Boolean("Programmer.BlindActive",
true)</step> <!-- go blind -->
<step>Programmer.SetBlindMode(false, 0)</step>
<step>Attribute.Mask.IncludeAll("Palette")</step> <!-- ??? -->
<step>Group.RecallGroupNumeric(17)</step> <!-- recall group #17 -->
<step>Palette.ApplyPalette("Location=Positions,0,68", false)</step>
<!-- set to palette Pos 0,68 -->
<step>ActionScript.SetProperty("Palette.CurrentPaletteHandle",
handle:"Location=Positions,1,71")</step>
<step>Palette.StoreCurrentPaletteReplace()</step> <!-- store as
palette 1,71 -->
<step>Programmer.Editor.Clear(Attribute.Mask.Clear.Value,
Programmer.Editor.Fixtures.Clear.Presets,
false, Expert.ClearMenu.FadeTime)</step> <!-- clear -->
<step>Group.RecallGroupNumeric(17)</step> <!-- recall group #17 -->
<step>Palette.ApplyPalette("Location=Positions,0,69", false)</step>
<!-- set to palette Pos 0,69 -->
<step>ActionScript.SetProperty("Palette.CurrentPaletteHandle",
handle:"Location=Positions,1,72")</step>
<step>Palette.StoreCurrentPaletteReplace()</step> <!-- store as
palette 1,72 -->
<step>Programmer.Editor.Clear(Attribute.Mask.Clear.Value,
Programmer.Editor.Fixtures.Clear.Presets,
false, Expert.ClearMenu.FadeTime)</step> <!-- clear -->
<step>ActionScript.SetProperty.Boolean("Programmer.BlindActive",
false)</step> <!-- exit blind -->
<step>Programmer.SetBlindMode(false, 0)</step>
</sequence>
</macro>

<macro id="UpChaseMove" name="Up Chase Move">
<!-- This shifts palettes 1,60~69 in pairs. -->
<sequence>
<step>Handles.SetSourceHandleFromHandle(Handles.GetHandle("Positions",0,69))
</step>
<step>ActionScript.SetProperty.Enum("Handles.OperationMode",
"move")</step>
<step>Handles.CopyDestination("Positions", 89)</step>
<step>Handles.ClearSelection()</step>

<step>Handles.SetSourceHandleFromHandle(Handles.GetHandle("Positions",0,68))
</step>
<step>ActionScript.SetProperty.Enum("Handles.OperationMode",
"move")</step>
<step>Handles.CopyDestination("Positions", 88)</step>
<step>Handles.ClearSelection()</step>

<!-- add more pairs to shift palettes here, see full file
</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 multiple steps are briefly explained inline. It is best understood with the applications hints below:

- a chaser should be created which toggles fixture group #17 between palettes #71 und #72
- various palettes should be stored as palettes #60~69, with the even numbers being something like 'foreground' and the odd numbers something like 'background'
- the macro Up Chase Move cycles through these palettes in pairs, in a way that palette #68 always contains a 'foreground' palette and #69 is always the corresponding 'background' palette. You may apply this macro multiple times. The functions used to move a palette are similar to [moving a chase handle](#).
- finally the macro Up Two Change toggles to blind, copies the contents from palette #68 to #71 and #69 to #72, and exits blind mode. This can be looked at in [ColourChaseChanger \(V2\)](#).

⇒ Jonas Nijs's ColourChaseChanger(v2) macro is very very useful. But it's a little uncomfortable to use in busking show, because it needs some step to copy palettes. So I made a macro which circulate 8 palettes with one button. If you want to circulate 10 or more, you can edit it.

How to use it

1. [make this macro available](#)
2. Using fixture group #17, create a chaser which oscillates between position palette 1,71 and 1,72.
3. Fire UpChaseMove (possibly a few times) so select another pair of palettes.
4. Fire UpTwoColorChange to make these palettes used in the chase.

From:

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

Permanent link:

https://www.avosupport.de/wiki/macros/example/palettechasechanger_v3?rev=1535389387

Last update: **2018/08/27 17:03**

