

Example

## ColourChaseChanger (V2)

<b>by:</b>	Jonas Nijs, Dec. 2017
<b>published:</b>	December 2017
<b>description:</b>	changes the colours of a color chase to any color you want
<b>remarks:</b>	This is an updated version <a href="#">to the old one</a> , this time using system syntax so it goes faster and whitout your screen flickering.
	Also see the author's remarks on how to use it ( <a href="#">below the code</a> )
	<a href="#">In the original post</a> there is also a short manual linked which you might find helpful.

Kim Wida developed a slightly changed version: [PaletteChaseChanger \(V3\)](#).

[change](#), [chase](#), [blind](#), [colour](#)

### functions

- [ActionScript.SetProperty.Boolean](#)
- [Programmer.SetBlindMode](#)
- [Group.RecallGroupNumeric](#)
- [Palette.ApplyPalette](#)
- [ActionScript.SetProperty](#)
- [Palette.StoreCurrentPaletteReplace](#) - alternatively [Palette.MergeCurrentPalette](#) (see [explanation](#))
- [Programmer.Editor.Clear](#)

### properties

- [Programmer.BlindActive](#)
- [Palette.CurrentPaletteHandle](#)
- [Attribute.Mask.Clear.Value](#)
- [Programmer.Editor.Fixtures.Clear.Presets](#)
- [Expert.ClearMenu.FadeTime](#)

## Code

[ColorChaseChanger\\_v2.xml](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<avolites.macros>
<!-- V2.0 by Nijs Jonas 7/12/2017 -->

<macro id="UserMacro.colchasechanger2">
  <name>Color chase changer</name>
  <sequence>
    <step
```

```
pause="0.01">ActionScript.SetProperty.Boolean("Programmer.BlindActive",
true)</step>
  <step pause="0.01">Programmer.SetBlindMode(false, 0)</step>
  <step pause="0.01">Group.RecallGroupNumeric(100)</step>
  <step pause="0.01">Palette.ApplyPalette("Location=Colours,1,16",
false)</step>
  <step
pause="0.01">ActionScript.SetProperty("Palette.CurrentPaletteHandle",
handle:"Location=Colours,2,1")</step>
  <step pause="0.01">Palette.StoreCurrentPaletteReplace()</step>
  <step
pause="0.01">Programmer.Editor.Clear(Attribute.Mask.Clear.Value,
Programmer.Editor.Fixtures.Clear.Presets, false,
Expert.ClearMenu.FadeTime)</step>
  <step pause="0.01">Group.RecallGroupNumeric(100)</step>
  <step pause="0.01">Palette.ApplyPalette("Location=Colours,1,17",
false)</step>
  <step
pause="0.01">ActionScript.SetProperty("Palette.CurrentPaletteHandle",
handle:"Location=Colours,2,2")</step>
  <step pause="0.01">Palette.StoreCurrentPaletteReplace()</step>
  <step
pause="0.01">Programmer.Editor.Clear(Attribute.Mask.Clear.Value,
Programmer.Editor.Fixtures.Clear.Presets, false,
Expert.ClearMenu.FadeTime)</step>
  <step
pause="0.01">ActionScript.SetProperty.Boolean("Programmer.BlindActive",
false)</step>
  <step pause="0.01">Programmer.SetBlindMode(false, 0)</step>
</sequence>
</macro>
</avolites.macros>
```

## Explanation

a brief explanation of the syntax used. For all the other XML details please refer to [Formats and syntax](#)

Essentially this macro does:

- enter blind mode
- replace a palette - e.g. our foreground colour - for a certain fixture group with another palette, clear
- replace yet another palette - e.g. our background colour - for a certain fixture group with another palette, clear
- exit blind mode

The commands are as follows:

- `ActionScript.SetProperty.Boolean("Programmer.BlindActive", true)` and `Programmer.SetBlindMode(false, 0)` toggle to blind mode, see [Blind Mode On/Off](#)
- `Group.RecallGroupNumeric(100)` recalls a specific group of fixtures - here group no. 100
- `Palette.ApplyPalette("Location=Colours,1,16", false)` selects the colour palette on the first page of the Colours workspace window, 16th slot
- `ActionScript.SetProperty("Palette.CurrentPaletteHandle", handle:"Location=Colours,2,1")` sets the colour palette on the second page on the 1th slot into the desks memory, i.e. makes it the current palette handle for the next actions
- `Palette.StoreCurrentPaletteReplace()` replaces the palette in the desks memory - the active palette handle - with what's in the programmer  
**HINT:** if you instead want to replace only a part of the palette then you need to merge it instead of replace the entire palette. Use [Palette.MergeCurrentPalette](#) instead
- `Programmer.Editor.Clear(...)` clears the programmer (for the parameters see [Programmer.Editor.Clear](#) - clears according to the current clear mask, clearing the presets as setglobally, in the globally set clear/release time)
- repeat the above steps, now with `Palette.ApplyPalette("Location=Colours,1,17", false)` and `ActionScript.SetProperty("Palette.CurrentPaletteHandle", handle:"Location=Colours,2,2")`, in order to replace the background colour
- `ActionScript.SetProperty.Boolean("Programmer.BlindActive", false)` and `Programmer.SetBlindMode(false, 0)` exit blind mode again - see above

## How to use it

1. [make this macro available](#)
2. make a color chase that uses color pallets
3. place those color pallets on the color pallets windows page 2 positions 1 and 2
4. create a group for all the fixtures used in that color chase and give that group usernumber 100
5. when you want to change the colors, place the 2 new colors you want on the color pallets page 1 positions 16 and 17
6. run the macro and have fun.

From:

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

Permanent link:

[https://www.avosupport.de/wiki/macros/example/colourchasechanger\\_v2](https://www.avosupport.de/wiki/macros/example/colourchasechanger_v2)

Last update: **2020/04/05 08:02**

