

Example

Set Lee Colour

by:	Sebastian Beutel
published:	June 2025, here
description:	set numeric lee colour on selected fixtures and fade this in over a time
remarks:	from https://www.facebook.com/groups/Avolites/posts/3500126146786261

[LEE](#), [colours](#), [blind](#), [group](#)

In order to accomplish that we abuse [Palette.Numeric.InputValue](#), create an arbitrary group, and fade out from blind mode.

functions

- [Group.CheckIfGroupExists](#)
- [Group.ReplaceGroupOnHandle](#)
- [Group.StoreGroup](#)
- [ActionScript.SetProperty.Boolean](#)
- [Programmer.SetBlindMode](#)
- [Group.RecallGroupNumeric](#)
- [Colour.ApplyColourFilterByIndex](#)

affected properties

- [Group.Numeric.IsValid](#)
- [Programmer.BlindActive](#)
- [Palette.Numeric.InputValue](#)

control structures

- [step condition](#)

File with macros for 0, 1 and 3 seconds here:

lee.xml

Code

[setLeeColours.xml](#)

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

<!--
  Pascal Njienhuis
```

*https://www.facebook.com/groups/Avolites/posts/3500126146786261/
set LEE timed colour on selected fixtures in a certain fade time
coded: Sebastian Beutel, June 2025*

-->

```
<macro id="Wiki.Macros.SetColourSeletec.Blind.1" name="Set Lee Colour
in 1 sec">
  <sequence>
    <step>Group.CheckIfGroupExists("999",
"Group.Numeric.IsValid")</step>
    <step pause="0.05" condition="Math.IsEqual(Group.Numeric.IsValid,
True)">Group.ReplaceGroupOnHandle(userNumber:999)</step>
    <step pause="0.05" condition="Math.IsEqual(Group.Numeric.IsValid,
False)">Group.StoreGroup(userNumber:999)</step>
    <step
pause="0.01">ActionScript.SetProperty.Boolean("Programmer.BlindActive",
true)</step>
    <step pause="0.01">Programmer.SetBlindMode(false, 0)</step>
    <step>Group.RecallGroupNumeric(999)</step>
    <step>Colour.ApplyColourFilterByIndex(1,
int:Palette.Numeric.InputValue)</step>
    <step
pause="0.01">ActionScript.SetProperty.Boolean("Programmer.BlindActive",
false)</step>
    <step pause="0.01">Programmer.SetBlindMode(true, 1)</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](#)

- `Group.CheckIfGroupExists()` checks whether group 999 already exists
- if yes then `Group.ReplaceGroupOnHandle()` replaces the group with the currently selected fixtures
- if no then `Group.StoreGroup()` stores group 999 with the currently selected fixtures
- `ActionScript.SetProperty.Boolean("Programmer.BlindActive", true)` and `Programmer.SetBlindMode(false, 0)` enter the Blind mode (see [Blind Mode On/Off](#) for details)
- `Group.RecallGroupNumeric()` recalls the group with the currently selected fixtures (remember: Blind has its own programmer)
- `Colour.ApplyColourFilterByIndex()` applies the Lee colour based on the number which was set prior to firing the macro (you need to type a number **and then press ENTER or EXIT!**)
- we exit Blind mode again and `Programmer.SetBlindMode(true, 1)` makes sure it is faded to live

How to use it

1. make sure group 999 is nowhere used (or rewrite the macros to use another group number)
2. [make this macro available](#)
3. select some fixtures
4. type a number and **press ENTER** (or EXIT)
5. fire one of the macros

From:

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

Permanent link:

<https://avosupport.de/wiki/macros/example/setleecolour>

Last update: **2025/06/11 12:32**

