Example

Inhibit selected fixtures

by:	Sebastian Beutel
published:	here, October 2022
description:	inhibits selected fixtures: sets them @0% and freezes
remarks:	idea by John Richardson, see https://www.facebook.com/groups/Avolites/posts/2641102656021952/

See Inhibit selected fixtures - that freezes the entire fixtures while here we freeze only the dimmer attribute.

inhibit, set, dimmer, selected, freeze, clear

> The inhibit is just a quick way of dousing a lamp or lamps and then reinstating them without affecting anything in the programmer or within any cues. So I may be running an effect on some lamps and I just need to kill it for a second and then turn it back on without messing about with the programmer or cues.

The 'without messing with the programmer' part isn't possible: you need to select the fixtures which you want to inhibit before calling the macro – and selecting fixtures brings them into the programmer. There is no way to avoid this.

functions

- Programmer.Editor.Fixtures.IncrementDimmer
- Programmer.Editor.Selection.GetSelectedHandles
- Fixtures.Patch.FreezeFixtures
- Programmer.Editor.ClearAll

Code

InhibitSelectedFixtures_v2.xml

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

<!-- Macros to mimic an inhibit function: fixtures are set to 0% and
then frozen -->
<!-- idea by John Richardson -->
<!-- Sebastian Beutel, October 2022 -->
<!-- https://www.facebook.com/groups/Avolites/posts/2641102656021952/ -
->

<avolites.macros>

  <macro id="Wiki.Macros.InhibitSelectedFixturesDimmer" name="Inhibit
```

```
Selected Fixtures Dimmer">
    <description>Inhibit On.</description>
    <sequence>
      <step>Programmer.Editor.Fixtures.IncrementDimmer(-10000, 1.0,
true)</step>
<step>Programmer.Editor.Fixtures.SetContextAttributeFromId(16)</step>
<step>Programmer.Editor.Fixtures.SetSelectedControlsFrozen(true)</step>
      <step>Programmer.Editor.ClearAll(false, false)</step>
    </sequence>
  </macro>

  <macro id="Avolites.Macros.UninhibitSelectedFixtures" name="Uninhibit
Selected Fixtures">
    <description>Inhibit Off</description>
    <sequence>
<step>Programmer.Editor.Fixtures.SetContextAttributeFromId(16)</step>
<step>Programmer.Editor.Fixtures.SetSelectedControlsFrozen(false)</step
>
      <step>Programmer.Editor.ClearAll(false, false)</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 macro `Inhibit Selected Fixtures` sets the currently selected fixtures' dimmer at 0% and freezes them:

- `Programmer.Editor.Fixtures.IncrementDimmer` sets the dimmer at 0
- `Programmer.Editor.Selection.GetSelectedHandles(“Windows.PatchView.Handles”)` puts the current selection into a property
- `Programmer.Editor.Fixtures.Patch.FreezeFixtures(Windows.PatchView.Handles, True)` freezes the fixtures which are in this property
- `Programmer.Editor.ClearAll(false, false)` clears the programmer

The second macro `Uninhibit Selected Fixtures` unfreezes the currently selected fixtures:

- `Programmer.Editor.Selection.GetSelectedHandles(“Windows.PatchView.Handles”)` puts the current selection into a property
- `Programmer.Editor.Fixtures.Patch.FreezeFixtures(Windows.PatchView.Handles, False)` unfreezes the fixtures which are in this property
- `Programmer.Editor.ClearAll(false, false)` clears the programmer

How to use it

1. make this macro available
2. in order to inhibit fixtures select them and fire the first macro
3. in order to uninhibit fixtures select them and fire the second macro