Hello,

Happy new year to you too. Sorry for the delay in getting back to you, releasing version 17.0 and other ongoing work was taking up my attention.

You can get hold of the inverted value of any fixture control using the `Programmer.Editor.Fixtures.Patch.IsFixtureControlInverted` function. If you give it a property it will save it to that property however you can also just use the return value. You are only getting the value of a single fixture so you would need to choose a context/main fixture to base your value on, unfortunately as handle types are not defined for that namespace (and it takes `FixtureHandle` instead of `Handle`) it is necessary to use the string and manually cast it to a handle.

Example:
```
Programmer.Editor.Fixtures.Patch.InvertFixtureControls("fixtureHandleUN=1", "Pan",
    !Programmer.Editor.Fixtures.Patch.IsFixtureControlInverted(handle:"fixtureHandleUN=1",
    "Pan", null))
```

Based on your code you could do this:
```
Programmer.Editor.Selection.GetSelectedHandles("Windows.PatchView.Handles")
Programmer.Editor.Fixtures.Patch.InvertFixtureControls(Windows.PatchView.Handles, "Pan",
    !Programmer.Editor.Fixtures.Patch.IsFixtureControlInverted(Handles.FirstOrDefault(Window
    s.PatchView.Handles), "Pan", null))
```

That is two lines above, the second is particularly long so may have wrapped, it might even let you do it in two lines in the macro as long as they are within the same step as I would expect it to ignore whitespace. There is a complementary `Handles.LastOrDefault` that you could also use to get one of the handles or alternatively you could use `Programmer.Editor.Selection.ContextFixtureHandle` property for this. The control name can be any of the predefined attribute ID strings (i.e. what you would get in the personality file without spaces) or it can be the integer control ID for any other (non-predefined) attribute.

There might be other ways of going about this but this was the first one I tried based your example and depending on where you get your selection from has the potential benefit of not requiring any other intermediate variables or context to use it.

Hope this helps.

Gregory Haynes
--
**Gregory Haynes**
**Senior Developer**
**Avolites Limited**

[www.avolites.com](www.avolites.com)

Hi Gregory,

I hope you had a nice festive season – I wish you all the best for a fab new year 😊

Macro question of the day: how can I create a 'toggle invert' macro which toggles e.g. Pan for the selected fixtures?

I did distinct invert and uninvert macros which work well, like this:

```
<macro id="Avolites.Macros.InvertPanSB" name="05 Invert Pan SB">
  <sequence>
    <step>Programmer.Editor.Selection.GetSelectedHandles("Windows.PatchView.Handles")</step>
    <step>Programmer.Editor.Fixtures.Patch.InvertFixtureControls(Windows.PatchView.Handles, "Pan", True)</step>
  </sequence>
</macro>

<macro id="Avolites.Macros.UnInvertPanSB" name="05 UnInvert Pan SB">
  <sequence>
    <step>Programmer.Editor.Selection.GetSelectedHandles("Windows.PatchView.Handles")</step>
    <step>Programmer.Editor.Fixtures.Patch.InvertFixtureControls(Windows.PatchView.Handles, "Pan", False)</step>
  </sequence>
</macro>
```

However, is there a way to toggle the invert property? I did find various functions
(Programmer.Editor.Fixtures.Patch.ToggleAttributeInvert(),
Programmer.Editor.Fixtures.Patch.ToggleContextAttributeInvert()) but I fail to set the environment, e.g. set context attributes, or refer this to the selected fixtures (if this is possible at all). May I ask for another little push in the right direction?

Thanks in advance. Sebastian