

Example

# Patch - Park fixtures 1 thru 100000

<b>by:</b>	Sebastian Beutel, July 2020
<b>idea:</b>	Yenz Nyholm, see <a href="https://www.facebook.com/groups/Avolites/permalink/1931175330348025/">https://www.facebook.com/groups/Avolites/permalink/1931175330348025/</a>
<b>description:</b>	Parks/unparks fixtures 1 ~ 100000

[park, fixtures](#)

## functions

- [Menu.Stack.PushOrReloadMenu](#)
- [Menu.InjectInput](#)
- [Programmer.Editor.Selection.GetSelectedHandles](#)
- [Handles.SetSourceHandleRangeFromHandles](#)
- [Patch.Repatch.SetDmxAddressesFromSelectedFixture](#)
- [Patch.Repatch.ParkSelectedFixtures](#)
- [Handles.ClearSelection](#)
- [Programmer.Editor.ClearAll](#)

## affected properties

- [Windows.PatchView.Handles](#)
- [Patch.Repatch.SelectedFixtureHandles](#)

The repatch menu does not actually use the PatchSelection, it has its own property Patch.Repatch.SelectedFixtureHandles.

## Code

[ParkFixtures1Thru100000.xml](#)

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

<!--
  Sebastian Beutel, July 2020
  idea by Yenz Nyholm
  see
  https://www.facebook.com/groups/Avolites/permalink/1931175330348025/
-->

<macro id="Avolites.Macros.ParkFixturesOneThru100000" name="Park
Fixtures 1 Thru 100000">
  <description>Parks fixtures 1 ~ 100000.</description>
```

```
<sequence>
  <step pause="0.001">Menu.Stack.PushOrReloadMenu("Primary",
"Expert.Root.Program")</step>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "Channel", "", 0)</step>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 1)</step
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "KeypadThrough", "", 0)</st
ep>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 1)</step
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 0)</step
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 0)</step
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 0)</step
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "KeypadEnter", "", 0)</step
>
  <step
pause="0.01">Programmer.Editor.Selection.GetSelectedHandles("Windows.Pa
tchView.Handles")</step>
  <step
pause="0.01">Handles.SetSourceHandleRangeFromHandles(Windows.PatchView.
Handles, true)</step>
  <step
pause="0.01">Patch.Repatch.SetDmxAddressesFromSelectedFixture()</step>
  <step pause="0.01">Patch.Repatch.ParkSelectedFixtures()</step>
  <step pause="0.01">Handles.ClearSelection()</step>
  <step pause="0.01">Programmer.Editor.ClearAll()</step>
</sequence>
</macro>

<macro id="Avolites.Macros.UnparkFixturesOneThru100000" name="Unpark
Fixtures 1 Thru 100000">
  <description>Unparks fixtures 1 ~ 100000.</description>
```

```
<sequence>
  <step pause="0.001">Menu.Stack.PushOrReloadMenu("Primary",
"Expert.Root.Program")</step>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "Channel", "", 0)</step>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 1)</step>
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "KeypadThrough", "", 0)</st
ep>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 1)</step>
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 0)</step>
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 0)</step>
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "NumericKeys", "", 0)</step>
>
  <step
pause="0.001">Menu.InjectInput("OnButtonDown", "KeypadEnter", "", 0)</step>
>
  <step
pause="0.01">Programmer.Editor.Selection.GetSelectedHandles("Windows.Pa
tchView.Handles")</step>
  <step
pause="0.01">Handles.SetSourceHandleRangeFromHandles(Windows.PatchView.
Handles, true)</step>
  <step
pause="0.01">Patch.Repatch.SetDmxAddressesFromSelectedFixture()</step>
  <step
pause="0.01">Patch.Repatch.UnParkSelectedFixtures(true)</step>
  <step pause="0.01">Handles.ClearSelection()</step>
  <step pause="0.01">Programmer.Editor.ClearAll()</step>
</sequence>
</macro>

</avolites.macros>
```

Here, the pause="0.01" seems to be required - if there is no pause, always the previously for

repatching selected fixtures will be parked, and if there aren't any an exception is thrown.

## Explanation

This explains the functional steps within the sequence. For all the other XML details please refer to [Formats and syntax](#)

The following code is taken from [Patch - Park selected fixtures](#)

- The first line retrieves the current selection and stores it in a menu property, it is not important which one just that it is the correct type.
- From that we then set the source handle selection (this is the same selection that is used for copy, move, set legend etc.).
- The third line copies the source handle selection to the Repatch property (and initialises other properties used in the repatch menu).
- The fourth line performs the Park operation and
- the last line clears the source handle selection once you are done.

## How to use it

- [make this macro available](#)
- When needed, select some fixtures and apply this macro in order to park these fixtures.

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

Permanent link:  
<https://www.avosupport.de/wiki/macros/example/parkfixtures1thru100000?rev=1594386369>

Last update: **2020/07/10 13:06**

