

Example

# Set Shapes (and KFS) Direction

<b>by:</b>	Gregory Haynes, January 2019
<b>published:</b>	<a href="http://forum.avolites.com/viewtopic.php?f=20&amp;t=5961">http://forum.avolites.com/viewtopic.php?f=20&amp;t=5961</a>
<b>description:</b>	Set keyframe shapes direction
<b>remarks:</b>	

[keyframe](#), [shape](#), [direction](#)

The function that the Effects Editor window uses to change the direction is `Editor.Shapes.SetShapeDirection` however the parameters for this are the ID of the shape and the direction object that should be set. The object is not just a name but an item with various parameters which I'm not sure will be possible to access or create in macros. As a key frame shape is still a shape you should be able to use the standard shape functions to make changes but may mean that the UI does not update properly. You can select key frame shapes in the Shape Menu by pressing Edit followed by Select Shape and then use the Direction options within the shape menu.

## functions

- [Editor.Shapes.ProgrammerShapesList](#)
- [Editor.Shapes.SelectAllShapes](#)
- [Editor.Shapes.SetCurrentShapesDirection](#)

## affected properties

- [Editor.Shapes.ProgrammerShapes.Empty](#)
- [Editor.Shapes.EditShapesEmpty](#)

## control structures

- [Control Structures](#)

## Code

[shapedirection.xml](#)

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

  <macro id="Avolites.Macros.ProgrammerShapes.Direction.RightToLeft"
name="Programmer Shapes Right to Left">
  <description>Select all programmer shapes and set the direction to
Right to Left.</description>
  <sequence>
    <step>Editor.Shapes.ProgrammerShapesList()</step>
```

```
<step
condition="!Editor.Shapes.ProgrammerShapes.Empty">Editor.Shapes.SelectA
llShapes()</step>
<step
condition="!Editor.Shapes.EditShapesEmpty">Editor.Shapes.SetCurrentShap
esDirection("RightToLeft")</step>
</sequence>
</macro>

<macro id="Avolites.Macros.ProgrammerShapes.Direction.LeftToRight"
name="Programmer Shapes Left to Right">
<description>Select all programmer shapes and set the direction to
Left To Right.</description>
<sequence>
<step>Editor.Shapes.ProgrammerShapesList()</step>
<step
condition="!Editor.Shapes.ProgrammerShapes.Empty">Editor.Shapes.SelectA
llShapes()</step>
<step
condition="!Editor.Shapes.EditShapesEmpty">Editor.Shapes.SetCurrentShap
esDirection("LeftToRight")</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](#)

- `Editor.Shapes.ProgrammerShapesList` gets all the programmer shapes including key frame shapes
- `Editor.Shapes.SelectAllShapes` selects all the programmer shapes if there are any
- `Editor.Shapes.SetCurrentShapesDirection("...")` sets the direction of the selected shapes if there are any

## How to use it

This is an example for two shape directions - you may add more macros for other directions. Put the macros on a button to quickly set the direction you want.

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

Permanent link:  
<https://www.avosupport.de/wiki/macros/example/setshapedirection>

Last update: **2019/05/03 19:35**

