Active Binding

You will have noticed that the factory macros for palette fade time and palette overlap feature something special: if e.g. you fire the macro to set palette overlap to 50%, this macro handle (key or button) is highlighted as active until you set palette overlap to another value. In this case, this is very useful to always have an overview what values overlap and fade time are set to.

Having a look into the code reveals the secret:

```
  <macro name="Palette Overlap 50%" id="Avolites.Macros.PaletteOverlap50">
    <description>Set master palettes fade to 50%.</description>
    <active binding="{propertyLink id='Palette.MasterOverlap'
converter='Math.EqualityConverter' converterParameter='0.5'}"/>
    <sequence>
<step>ActionScript.SetProperty.Float("Palette.MasterOverlap",0.5)</step>
    </sequence>
  </macro>
```

Obviously the interesting line is this one:

```
<active binding="{propertyLink id='Palette.MasterOverlap'
converter='Math.EqualityConverter' converterParameter='0.5'}"/>
```

You could read this as

- bind the 'active'ness of this handle to a condition
- this condition is true (and the bound handle marked active) when
  - a property - in this case `Palette.MasterOverlap'`
  - equals (see `converter='Math.EqualityConverter'`
  - a certain value, see `converterParameter='0.5'` .

Another - simpler - way would be to directly link to a property which needs to be of type Boolean:

```
<active binding="Windows.Scrolling.ScrollLock" />
```

In any case, **you can only bind to properties, not to functions**.

```
The active binding requires there to be a property, properties fire events
when they change which allow the UI to update. Generally the UI does not
continually call functions to get current values it depends on events to
allow it to update. (Gregory Haynes)
```