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 are set to.

Having a look into the code receals the secret:

```xml
<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'}"/>
```

A reasonable assumption for this to understand is:

- 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'` .

At he moment, this is only listed here for information. There are a few potentially interesting aspects which might eventually be looked into:

- this is, next to `description` and `sequence`, another child of the `macro` element
- interesting syntax, with curly braces {} and propertyLink, converter, and converterParameter.

    Maybe we meet this again somewhere 🙂