

States and transitions basics QML vs. Silverlight

With QML if you want to implement states and transitions, you would basically define them separately. Basically you would start first by defining the states, and what properties would change when the state is changed, this could be handled for example like this:



```
states: [
  State {
    name: "pressed"
    PropertyChanges { target: button; color: "blue"}
  },
  State {
    name: "normal"
    PropertyChanges { target: button; color: "green"}
  }
]
```

Now basically if you would have for example a rectangle which Id is button, you could change its color by simply changing its state. The state can have multiple property changes, which can have different targets as well. With the state definition, the change would happen instantly, and to get the change animated, you would also need to add the transitions definitions. With transitions definition, you can define from and to states to define the exact situation where the transition will be used and well as the animations used in the transition. Example definition for the above states definition could look like this:

```
transitions: [
  Transition {
    from: "pressed"
    to: "normal"
    ColorAnimation { target: button; duration: 100}
  },
  Transition {
    from: "normal"
    to: "pressed"
    ColorAnimation { target: button; duration: 100}
  }
]
```

With Silverlight, you would need to handle the states rather differently, basically you would need to use `VisualStateManager` in which you could define transitions and states and animations all in same place.

Basically you would define a `VisualStateManager`, and inside it you would then define the named `VisualStates`, and inside each `VisualState` you would define `Storyboard` animations as needed.

```
<VisualStateManager.VisualStateGroups>
  <VisualStateGroup x:Name="Expressions">
    <VisualState x:Name="Normal">
      <Storyboard>
        <ColorAnimation
          Duration='00:00:01'
          To="Blue"
          Storyboard.TargetName="button"
          Storyboard.TargetProperty="(Control.Foreground).(SolidColorBrush.Color)">
        </Storyboard>
      </VisualState>
    <VisualState x:Name="pressed">
      <Storyboard>
        <ColorAnimation
          Duration='00:00:01'
          To="Green"
          Storyboard.TargetName="button"
          Storyboard.TargetProperty="(Control.Foreground).(SolidColorBrush.Color)">
        </Storyboard>
      </VisualState>
    </VisualStateGroup>
  </VisualStateManager.VisualStateGroups>
```

Then if any fine tuning would be needed you could also define `VisualTransitions` for each state separately inside the same `VisualStateGroup` you defined the states. There you can define the to and from states these transitions would take part in, as well as any additional fine tuning needed for the state change animation, here's quick example of `VisualTransition` implementation:

```
<VisualStateGroup.Transitions>
  <VisualTransition To="pressed" GeneratedDuration="00:00:02">
    <VisualTransition.GeneratedEasingFunction>
      <BounceEase Bounces="4" EasingMode="EaseIn" Bounciness="2"/>
    </VisualTransition.GeneratedEasingFunction>
  </VisualTransition>
</VisualStateGroup.Transitions>
```

Then to change the states, you would basically call `GoToState` inside the controls C# code:

```
VisualStateManager.GoToState(this, "pressed", true);
```

Examples

Simple states and transitions example for silverlight: [Media:Silver SimpleStates1.zip](#)

