How to add configuration support to a control

Slide.Show 2 offers several methods for configuring a control. The base level of configuration comes in the form of pure XAML and template binding. Additional levels of configuration are also supported via XML configuration and theme configuration.

Create your property

If you are using template binding, then create a dependency property. This allows you to pass through settings from a control to the UI within the template.

```
public double ScrollPathButtonWidth
{
        get { return (double)GetValue(ScrollPathButtonWidthProperty); }
        set { SetValue(ScrollPathButtonWidthProperty, value); }
}

public static readonly DependencyProperty ScrollPathButtonWidthProperty =
        DependencyProperty.Register(
                "ScrollPathButtonWidth",
                typeof(double),
                typeof(AlbumViewer),
                null);
```

Add the property to the appropriate options class

Configurable options are broken up into modules or groups. These groups are organized by classes. Find the appropriate class for the control to which you are adding the configurable property.

```
public class AlbumViewerOptions
{
        …
        public double ScrollButtonWidth { get; set; }
```

For non-nullable types, set the value to that of the DefaultOptions (if available) within the constructor.

```
public AlbumViewerOptions()
{
        if (DefaultOptions != null)
        {
                …
                ScrollButtonWidth = DefaultOptions.AlbumViewer.ScrollButtonWidth;
        }
}
```

Add the property assignment to the configuration themes

A configuration theme is a class that defines a set of option values.

```
public class LightTheme : ConfigurationProvider
{
        …
        protected override void GetConfiguration()
        {
                …
                Options.AlbumViewer = new Options.AlbumViewerOptions()
                {
                        …
                        ScrollButtonWidth = 10,
                };
```

Add the property assignment to the XmlConfigurationProvider

The XmlConfigurationProvider class reads XML configuration settings and populates the appropriate options.

```
if (xmlOptions.ContainsKey("AlbumViewer"))
{
        Dictionary<string, string> d = xmlOptions["AlbumViewer"];
        Options.AlbumViewer = new Options.AlbumViewerOptions()
        {
                …
                ScrollButtonWidth =
                        ParseDouble(d, "ScrollButtonWidth") ??
                        Options.AlbumViewer.ScrollButtonWidth,
        };
}
```

Add the property assignment to your control

Now that you have the configurable property available, it can be set in code. Typically, this is done within the constructor of the control.

```
public AlbumViewer()
{
        Options.AlbumViewerOptions options = Configuration.Options.AlbumViewer;

        …
        ScrollPathButtonWidth = options.ScrollButtonWidth;
}
```