How to write a SlideViewer control

Slide.Show 2 displays slide images in an ImageViewer control and videos in a VideoViewer control. If you wish to handle additional content, then you should create your own control that inherits from the SlideViewer base control.

Inherit from SlideViewer

The SlideViewer base class exposes the Slide and Source properties. The Slide property has a reference to a Slide object. The Source property has a reference to the slide source (as a URI).

```
[TemplatePart(Name = ImageViewer.ElementImageName, Type = typeof(Image))]
public class ImageViewer : SlideViewer
{
```

Create your templated control

Follow the standards involved in creating a templated control that inherits from the SlideViewer control.

Override the OnSourceChanged method

The OnSourceChanged method is called when a change is made to the Source property of the SlideViewer class. This is where your viewer should perform its display logic.

```
protected override void OnSourceChanged()
{
        if (ImageElement != null)
        {
                ImageElement.Source = (Source != null ? new BitmapImage(Source) : null);
                OnMediaCompleted();
        }

        base.OnSourceChanged();
}
```

In the above example, the OnMediaCompleted method is called to notify Slide.Show 2 that its display logic is completed and that the slide can be transitioned out when its time is up.

Override the OnContentResized method (optional)

When the application gets resized, the OnContentResized method is called. By default, it sets the control's width and height. If you have special sizing needs, you may want to use this method to perform the logic.

Add creation logic to SlideViewer

Slide.Show 2 uses the static Create method of the SlideViewer class to determine which type of SlideViewer to create. Within the Create method, the code identifies the extension of the slide source, and then returns the appropriate SlideViewer instance (e.g. an ImageViewer instance for image extensions).

```
public static SlideViewer Create(Slide slide)
{

        if (slide == null)
        {
                throw new ArgumentNullException("slide");
```

```
        }

        string ext = Path.GetExtension(slide.Source.OriginalString).ToLower();

        switch (ext)
        {
                // image extensions
                case ".jpg":
                case ".jpeg":
                case ".png":
                case ".wmf":
                case ".tiff":
                case ".gif":
                case ".exif":
                case ".emf":
                case ".bmp":
                        return new ImageViewer(slide);

                // video extensions
                case ".wmv":
                case ".wma":
                case ".mp3":
                        return new VideoViewer(slide);
        }

        return null;
}
```