1) Create a new SL 2.0 application named **Resizer**.
2) Select the application type you prefer.
3) Le's first add some content to be resized. We'll edit the page's XAML file and add a canvas element:

```
<Canvas Width="400" Height="300" Background="Yellow">
</Canvas>
```

4) To the canvas we'll add some shapes, starting with a rectangle:

```
<Rectangle Width="200" Height="100" Canvas.Left="10" Canvas.Top="10"
Fill="Blue"></Rectangle>
```

5) Then we'll add an ellipse:

```
<Ellipse Width="200" Height="100" Canvas.Left="40" Canvas.Top="150"
Fill="Red"></Ellipse>
```

6) Part of the secret is to now add a scale transform. Note we'll name it so we can access it later:

```
<Canvas.RenderTransform>
    <ScaleTransform x:Name="myScale"></ScaleTransform>
</Canvas.RenderTransform>
```

7) With some content to resize, let's now see how we actually make it resize when the browser window resizes. Open the App.xaml.cs file for editing and add these fields:

```
private double __currentWidth = 0.0;
private double __currentHeight = 0.0;
```

8) Then look for the startup event handler. Add this code there (note we'll write the *GetBrowserSize* method in a moment):

```
Size size = GetBroswerSize();
__currentHeight = size.Height;
__currentWidth = size.Width;
```

9) We'll next need to hook the browser's resize event. For that, let's use *AttachEvent*. Add this code to the startup event handler:

```
HtmlPage.Window.AttachEvent("resize", Application_Resize);
```

10) We'll also want to detach the resize event, so find the Exit event handler and add this code:

```
HtmlPage.Window.DetachEvent("resize", Application_Resize);
```

11) Now we need to add an event handler for the resize event. Add this method:

```
private void Application_Resize(object sender,
System.Windows.Browser.HtmlEventArgs e)
{
}
```

12) To the event handler, add this code to find the current browser window size:

```
Size size = GetBroswerSize();
```

13) Now we need to find the scale transform for our content. To do that, we'll add this code:

```
Page sl = this.RootVisual as Page;
ScaleTransform myScale = sl.FindName("myScale") as ScaleTransform;
```

14) Now we actually scale the content by doing some simple math. Add this code:

```
myScale.ScaleX *= size.Width / __currentWidth;
myScale.ScaleY *= size.Height / __currentHeight;
```

15) Finally, we need to save the now-current browser window size so we can scale the next time the window resizes. To do that, add this code:

```
__currentHeight = size.Height;
__currentWidth = size.Width;
```

16) Now the only question remaining is how we determine the current size of the browser window, keeping in mind we'll want to do this in a browser-independent way. To begin, add this method:

```
private Size GetBroswerSize()
{
}
```

17) We want the current browser size, but we'll have to get that using the properties exported by the browser itself. We don't have access to *BrowserHost*, for example, at least not in managed code. But then we get into differences in how browsers report the size of the window. Mozilla has a pair of properties—*innerWidth* and *innerHeight*—that are exposed by the window object. So let's look for those. If we have them, we're working with Mozilla. Add this code:

```
Size size = new Size();
if (null != HtmlPage.Window.GetProperty("innerWidth"))
{
}
```

18) If the *innerWidth* property isn't null, we can work with the Mozilla values. Therefore, add this code:

```
size.Width = (double)HtmlPage.Window.GetProperty("innerWidth"); // Mozilla
size.Height = (double)HtmlPage.Window.GetProperty("innerHeight");
```

19) Of course, if the *innerWidth* property is null, we're working with Internet Explorer. IE reports the browser window size through the Body object's *clientWidth* and *clientHeight* properties. So add an else condition to work with these values:

```
else
{
    size.Width = (double)HtmlPage.Document.Body.GetProperty("clientWidth");
// IE
    size.Height = (double)HtmlPage.Document.Body.GetProperty("clientWidth");
}
```

20) Finally, of course, we need to return the size we came up with:

```
return size;
```

21) Compile and test.