

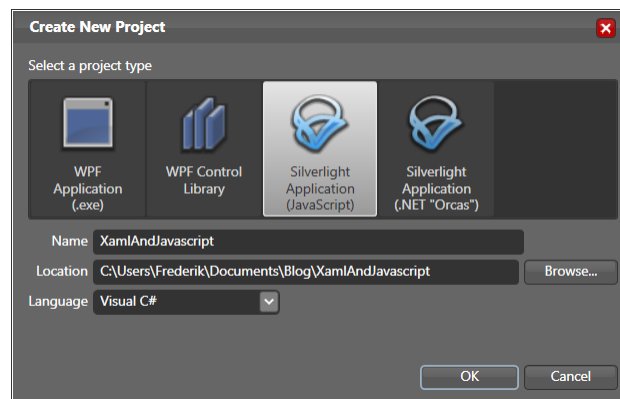
Xaml and javascript

Step 1: create a new project

I use Expression Blend 2.0 September preview for this tutorial. The preview can be downloaded from <http://www.microsoft.com/expression/products/download.aspx?key=blend>.

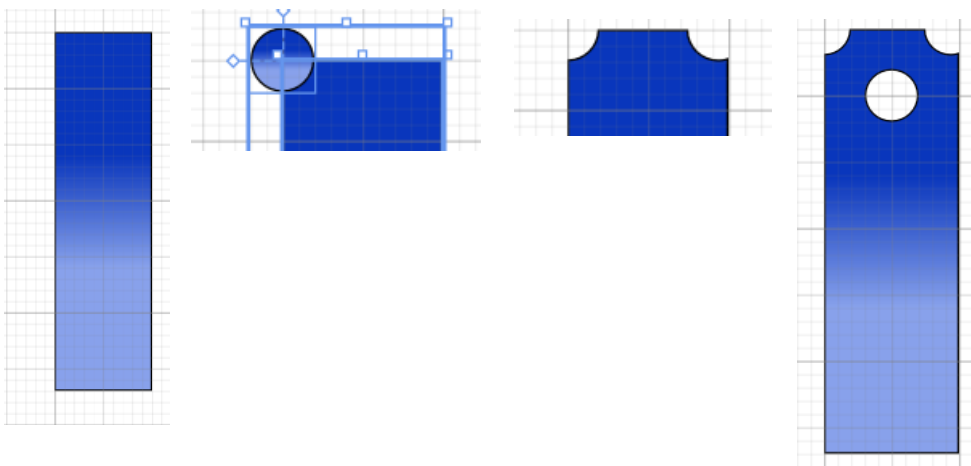
You also need a trial of Visual Studio 2008. You can download a 90 days trial form <http://www.microsoft.com/downloads/details.aspx?FamilyId=83C3A1EC-ED72-4A79-8961-25635DB0192B&displaylang=en>.

- Start Expression Blend and choose File → New → Project.
- Choose a Silverlight Application (Javascript) as template. Also choose a location for the project and give it a name.
- Set the width of the canvas Page to 240 and the height to 375.

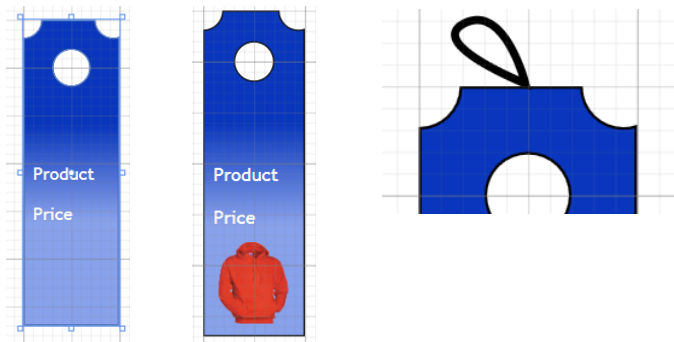


Step 2: create the layout of the label

- Draw a rectangle on the canvas page.
- Go to the properties panel and choose a gradient brush as fill color
- Create a gradient from dark blue (#FF0936BC) to light blue (#FF88A1EB)
- Select the Brush transform tool from the toolbox and rotate the gradient 90 degrees.
- Draw a circle on the canvas a place it on the upper left corner of the rectangle.
- Select the circle and rectangle and choose Object → Combine → Subtract
- Draw a circle in the upper right corner of the rectangle and perform the same operation.
- Finally draw a circle in the top middle of the rectangle and also subtract it form the rectangle.



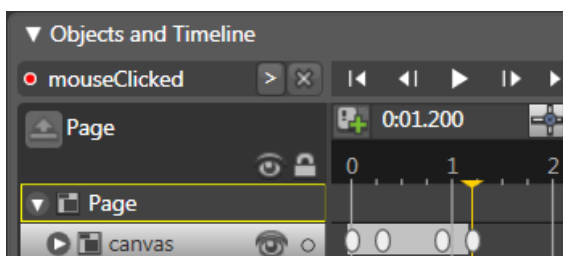
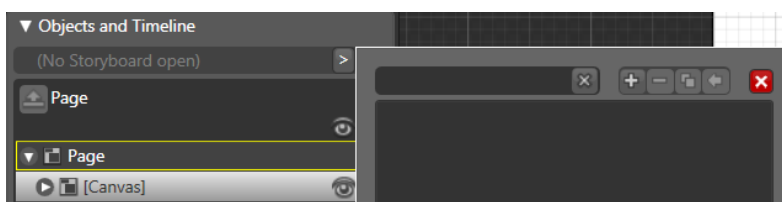
- Place a textblock on the rectangle, name it lblProduct, and choose white as forecolor and trebuchet as font.
- Place a second textblock on the rectangle with the same settings and name it lblPrice.
- Add a folder “Images” to the project and put the images pull.png and pants.png in the folder (right click → Add existing item)
- Place an image element at the bottom of the rectangle and choose the image pull or pants as source. Name the element image picProduct.
- Finally, choose the pen tool from the toolbox and draw a small holder for the label at the top of the rectangle. Set the fill to none and the stroke thickness to 3.



- Select all the elements on the canvas Page, right click on the selection and choose Group into → canvas. Name the canvas myLabel.
- Select the rotation point (white circle) of the canvas and drag it to the top of the rectangle.

Step 3: create the storyboards for the label

- Create a new storyboard in the objects and timeline panel. Name the storyboard “mouseClicked” and make sure the checkbox “create as resource” is checked because we will start the storyboard from code behind with javascript.
- Record a new keyframe at 0 seconds
- Move the playhead to 0:00:300 seconds, and rotate the canvas -10 degrees (use the transform section in the property panel to rotate the canvas)
- Move the playhead to 0:00:900 seconds and rotate the canvas 10 degrees.
- Finally, move the playhead to 0:01:200 and set the rotation of the canvas to 0.



- Create a second storyboard and name it mouseEntered.
- Record a new keyframe at 0 seconds
- Move the playhead to 0:00:300 seconds and place a new keyframe. Set the scale for X and Y to 1.1 in the transform section of the property panel.
- Right click the name of the storyboard in the objects and timeline panel and choose Duplicate.
- Right click the name of the storyboard again and choose Reverse.
- Right click the name of the storyboard one more time and choose Rename.
- Name the new storyboard mouseLeft.

Step 4: start the animations with javascript

- Save the project in Expression Blend and open it with Visual Studio 2008.
- Open the file Page.xaml and you will see all the XAML code that Blend has created: the 3 storyboards and all the elements for the label.
- Add the eventhandlers to start the storyboards on the canvas myLabel:

```
<Canvas Width="80" Height="281.784" Canvas.Left="80" Canvas.Top="14.216"
RenderTransformOrigin="0.5,0.12" x:Name="canvas" MouseEnter="Bigger"
MouseLeave="Smaller" MouseLeftButtonDown="startRotation">
```

- Open the file Page.xaml.js and add this code at the bottom of the page.

```
function Bigger(sender, EventArgs)
{
    sender.getHost().content.findName("mouseEntered").Begin();
}

function Smaller(sender, EventArgs)
{
    sender.getHost().content.findName("mouseLeft").Begin();
}

function startRotation(sender, EventArgs)
{
    sender.getHost().content.findName("mouseClicked").Begin();
}
```

Sender.getHost() addresses the silverlight component. The entire code searches for the storyboard with the name mouseEntered, mouseLeft or mouseClicked. After the storyboard is found, it will be started.

Step 5: interaction between HTML and Silverlight

Open Default.html and add a form below the div that contains the Silverlight element. Of course you can add style sheet to create a nice layout for this form.

```
<div class="settings">
<form method="post" action="">
  <div class="part1">
    <label>Product:</label><input type="text" id="product" />
    <br />
    <label>Price:</label><input type="text" id="price" />
  </div>
  <div class="part2">
    <input type="radio" name="picture" value="pull" checked="checked"/>
    <label>pull</label><br />
    <input type="radio" name="picture" value="pants"/>
    <label>pants</label>
    <br /><br />
    <input type="button" value="Create label" onClick="createLabel();" />
  </div>
</form>
</div>
```

When the button is clicked, the function createLabel will be executed. We will create this function in the file Page.xaml.js.

```
function createLabel()
{
  // get all the values from the HTML form
  var price = document.getElementById("price").value;
  var product = document.getElementById("product").value;
  var picture = document.getElementsByName("picture");
  // test which radiobutton is checked
  for(i=0;i<picture.length;i++)
  {
    if(picture[i].checked == true)
    {
      // create a variable that contains the path to the selected
      image
      var sSource = "Images/" + picture[i].value + ".png";
    }
  }

  // address the silverlight control. The id for the control is set in
  the function createSilverlight
  // this function can be found in the file Default.html.js
  var control = document.getElementById("SilverlightControl");

  // search the textbock elements and change the text property
  control.content.findName("lblProduct").Text = product;
  control.content.findName("lblPrice").Text = price;
  // search the image element and change the source property
  control.content.findName("picProduct").Source = sSource;
}
```

Press F5 to test the project, fill in the form and click the button.