

Course IPSL: Image processing with Silverlight

Chapter C1: The Binary Image Project

Copyright © by V. Miszalok, last update: 10-06-2010

Install 1) [Visual Web Developer 2010 Express English](#)
and 2) [Web Platform](#).

Preliminaries

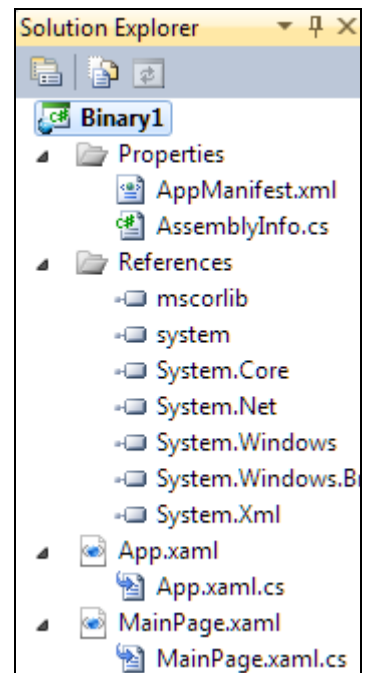
Guidance for **Visual Web Developer 2010 Express**:

- 1) Main Menu after start of VWD Express: Tools → Options → check lower left checkbox: Show all Settings → Projects and Solutions → Projects location: → C:\temp. → Text Editor (double click) → All Languages (double click) → Tabs → Indenting: None → Tab size: 2 → Insert spaces. → Text Editor (double click) → C# (double click) → Formatting → uncheck all three check boxes → OK. → Text Editor (double click) → XAML (double click) → Tabs → Indenting: None → Tab size: 1, Indent size: 1 → Insert spaces. → Text Editor (double click) → XAML (double click) → Formatting → uncheck all Auto-Formatting Events → OK.

- 2) Main Menu after start of VWD Express: File → New Project... → Installed templates: Visual C# (double click) → Silverlight → Silverlight Application
Name: Binary1 → Location: C:\temp → Create directory for solution: switch off → OK.
A New Silverlight Application-Window appears.
Uncheck "Host the Silverlight application in a new Web site".
Silverlight Version: Silverlight 4 → OK.
Start the default code by clicking Debug → Start Debugging F5. Binary1 - Windows Explorer will appear and show an empty
C:\temp\Binary1\Bin\Debug\Binary1TestPage.html.
Now you have to store 3 images into the directory C:\temp\Binary1\Bin\Debug\
www.miszalok.de/Images/Lena256.png Format 256x256
www.miszalok.de/Images/Butterfly.jpg Format 295x250
www.miszalok.de/Images/Ferropolis.jpg Format 1200x1600.

If the Solution Explorer - Binary1 window doesn't appear as shown on the right, you must open it manually: View → Other Windows → Solution Explorer.

In the Solution Explorer - Binary1 window click the branch MainPage.xaml. A split window will appear containing a empty rectangle above the default XAML code. Hide the rectangle by shifting up the horizontal splitter line to the upper rim. Delete the red and blue XAML code completely and replace it by the code of **Version 01: MainPage.xaml** which follows.



Version 01: MainPage.xaml

```
<UserControl x:Class="Binary1.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Viewbox>
    <Image Source="Lena256.png"/>
  </Viewbox>
</UserControl>
```

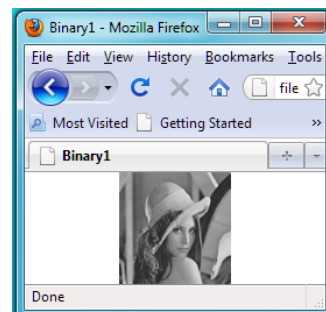
Start this code by clicking Debug → Start Debugging F5. Binary1 - Windows Explorer will appear and show

C:\temp\Binary1\Bin\Debug\Binary1TestPage.html with Lena256.png.

Experiments:

1. Try out the other images by changing the line

```
<Image Source="Lena256.png"/> to:
<Image Source="Butterfly.jpg"/>
<Image Source="Ferropolis.jpg"/>.
```
2. Drag the borders of Binary1 - Windows Explorer and observe how the image size follows the window size (Such an adaptive size behavior is due to the Viewbox-object containing the image. It adapts any content automatically to all displays (e.g. mobile phones).



Version 02: Binary1TestPage.html

Open C:\temp\Binary1\Bin\Debug\Binary1TestPage.html with a text editor and replace its default content by the following HTML-code and save it:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head>
    <title>Binary1</title>
    <style type="text/css">
      html, body { height: 100%; overflow: auto; }
      body { padding: 0; margin: 0; }
      #silverlightControlHost { height: 100%; text-align:center; }
    </style>
  </head>
  <body>
    <center>HTML in front of the Silverlight object.</center>
    <form id="form1" runat="server" style="height:70%">
      <object data="data:application/x-silverlight-2,"
        type="application/x-silverlight-2" width="100%"
        height="100%">
        <param name="source" value="Binary1.xap"/>
        <param name="minRuntimeVersion" value="4.0.50401.0" />
        <param name="autoUpgrade" value="true" />
        <a href
          ="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.50401.0"
          style="text-decoration:none">
          
          </a>
        </object>
      </form>
    <center>HTML below the Silverlight object.</center>
  </body>
</html>
```

Double click

C:\temp\Binary1\Bin\Debug\ Binary1TestPage.html to render it in your favorite browser:



Please notice:

1. Both lines of text do not resize when resizing the browser because they live outside of the Silverlight Viewbox object.
2. The Silverlight object loads a file named Binary1.xap which obviously contains any code.
3. The <a ...>-clause loads and installs your browser's Silverlight 4 plugin if there isn't.

Version 03: MainPage.xaml.cs

At first we have to add a name and an event handler to our Image.

```
Open MainPage.xaml and add a name and an event handler to <Image
Source="Lena256.png"/>:
<Image x:Name="myImage" Source="Lena256.png"
Loaded="myImage_Loaded"/>
```

Now open MainPage.xaml.cs and replace its default code by:

```
using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media.Imaging;

namespace Binary1
{
    public partial class MainPage : UserControl
    {
        WriteableBitmap wb0, wb1;
        Byte threshold = 128;
        public MainPage()
        {
            InitializeComponent();
        }
        private void myImage_Loaded( object sender, RoutedEventArgs e )
        {
            wb0 = new WriteableBitmap( (BitmapSource)myImage.Source );
            wb1 = new WriteableBitmap( (BitmapSource)myImage.Source );
            for ( int i=0; i < wb0.PixelWidth*wb0.PixelHeight; i++ )
            {
                if ( (Byte)(wb0.Pixels[i] & 0xff) < threshold ) //red channel only
                    unchecked { wb1.Pixels[i] = (Int32)0xff000000; } //black
                else unchecked { wb1.Pixels[i] = (Int32)0xffffffff; } //white
            }
            wb1.Invalidate();
            myImage.Source = wb1;
        }
    }
}
```



Please notice:
Mozilla Firefox, Google Chrome and others are 32-bit software. When you develop with Visual Web Developer 64 bit under Windows 7 64-bit OS you need a 64 bit browser (such as the 64 bit IE 8 and 9). All 32 bit browsers will just display the original Lena256.png.

Version 04: Sliding Threshold

MainPage.xaml:

```
<UserControl x:Class="Binary1.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
<Viewbox>
<StackPanel Orientation="Horizontal">
<Slider x:Name="mySlider" Width="100" Height="20" Maximum="255"
Value="128" ValueChanged="mySlider_ValueChanged"/>
<Image x:Name="myImage" Source="Lena256.png" Loaded="myImage_Loaded"/>
</StackPanel>
</Viewbox>
</UserControl>
```

MainPage.xaml.cs:

```
using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media.Imaging;
namespace Binary1
{
    public partial class MainPage : UserControl
    {
        WriteableBitmap wb0, wb1;
        Byte threshold = 128;
        public MainPage()
        {
            InitializeComponent();
        }
        private void myImage_Loaded( object sender, RoutedEventArgs e )
        {
            wb0 = new WriteableBitmap( (BitmapSource)myImage.Source );
            wb1 = new WriteableBitmap( (BitmapSource)myImage.Source );
            mySlider_ValueChanged( null, null );
        }
        private void mySlider_ValueChanged( object sender,
            RoutedEventArgs e )
        {
            if ( mySlider == null ) return;
            threshold = (Byte)mySlider.Value;
            for ( int i=0; i < wb0.PixelWidth*wb0.PixelHeight; i++ )
            {
                if ( (Byte)(wb0.Pixels[i] & 0xff) < threshold ) //red channel only
                    unchecked { wb1.Pixels[i] = (Int32)0xff000000; } //black
                else unchecked { wb1.Pixels[i] = (Int32)0xffffffff; } //white
            }
            wb1.Invalidate();
            myImage.Source = wb1;
        }
    }
}
```



Version 05: Borders, Textblock, Button

MainPage.xaml:

```
<UserControl x:Class="Binary1.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Viewbox>
    <Border BorderBrush="Black" BorderThickness="1" Margin="3">
      <StackPanel Orientation="Horizontal">
        <Slider x:Name="mySlider" Width="100" Height="20" Maximum="255"
          Value="128" ValueChanged="mySlider_ValueChanged"/>
        <Border BorderBrush="Black" BorderThickness="2"
          Width="25" Height="20" Margin="3">
          <TextBlock x:Name="myTextBlock" TextAlignment="Center"/>
        </Border>
        <Border BorderBrush="red" BorderThickness="1" Margin="3">
          <Image x:Name="myImage" Source="Lena256.png" Loaded="myImage_Loaded"/>
        </Border>
        <Button x:Name="myButton" Content="Reset" Width="40" Height="20"
          Margin="3" Click="myButton_Click"/>
      </StackPanel>
    </Border>
  </Viewbox>
</UserControl>
```

MainPage.xaml.cs:

Insert the following line below `threshold = (Byte)mySlider.Value;`
`myTextBlock.Text = threshold.ToString();`

Insert the following event handler below the existing code but above of both ending clauses:
`private void myButton_Click(object sender, RoutedEventArgs e)`
`{ myImage.Source = wb0;`
`}`

Insert the Silverlight content in an arbitrary HTML-file

Switch the compiler to produce a release-mode XAP-file via the main menu after start of VWD 2010 Express:
 Debug → Build Binary1.

Your `C:\temp\Binary1\bin\Release-directory` will now contain a XAP-file `Binary1.xap` that can be incorporated into any HTML-page by:

1. Store it in the same directory as the housing HTML-page.
2. Call it at any line from inside the `<body></body>` tags of the HTML-page by inserting the following HTML-code:

```
<form id="form1" runat="server" style="height:100%">
  <object data="data:application/x-silverlight-2,"
    type="application/x-silverlight-2" width="100%" height="100%">
    <param name="source" value="Binary1.xap"/>
    <param name="minRuntimeVersion" value="4.0.50401.0" />
    <param name="autoUpgrade" value="true" />
    <a href ="http://go.microsoft.com/fwlink/?LinkId=149156&v=4.0.50401.0"
      style="text-decoration:none">
      
    </a>
  </object>
</form>
```

Out of Browser = OOB Application

It is possible to detach `Binary1.xap` from its embedding `Binary1TestPage.html` and to create a stand-alone program that can be started in its own window.

This mechanism works with all platforms and all browsers running the Silverlight 4 plug-in.

1. Main menu after start of VWD 2010 Express: `Project` → `Binary1 Properties` → `Silverlight` → Check the checkbox: `Enable running application out of the browser.`
Click the button `Out-of-Browser Settings ...`. An `Out-of-Browser Settings-Window` appears.
Set `Width` to `400` and `Height` to `280`. Check `Use GPU Acceleration` and `Show install menu` → `OK` → Click `Debug` → `Build Binary1.`
Directory `C:\temp\Binary1\Bin\` will contain a new sub-directory `Release.`
Copy `Lena256.png`, `Butterfly.jpg` and `Ferropolis.jpg` into this new sub-directory `Release.`



Double click `Binary1TestPage.html` and the Silverlight application is rendered by the favorite browser.

Right click the application and a context menu will appear.

Click `Install Binary1 Application onto this computer...`

An `Install application window` asks where to install the shortcut.

Uncheck `Start menu` and check `Desktop` → `OK.`

Our program now starts a window: `Binary1 Application - localhost` and at the same time a persistent new icon named: `Binary1 Application` will appear on the desktop.

From now our application behaves as any normal application.

Whenever you click it, you will obtain a new

`Binary1 Application - localhost-window.`

Its home is the

`C:\Users\YourUserName\AppData\LocalLow\Microsoft\Silverlight\OutOfBrowser-directory.`

To uninstall `Binary1 Application - localhost` start `Binary1TestPage.html` again.

Right click the application and a context menu will appear.

Click `Remove this application....`