

# Windows Phone 7 Development

31 Days of Windows Phone 7 Development



[WWW.F5DEBUG.NET](http://WWW.F5DEBUG.NET)

February 1, 2012

Karthikeyan Anbarasan, Microsoft MVP

# Learn Windows Phone 7 Development in 31 Days – Day 21 – Working with BING Map Control in WP7

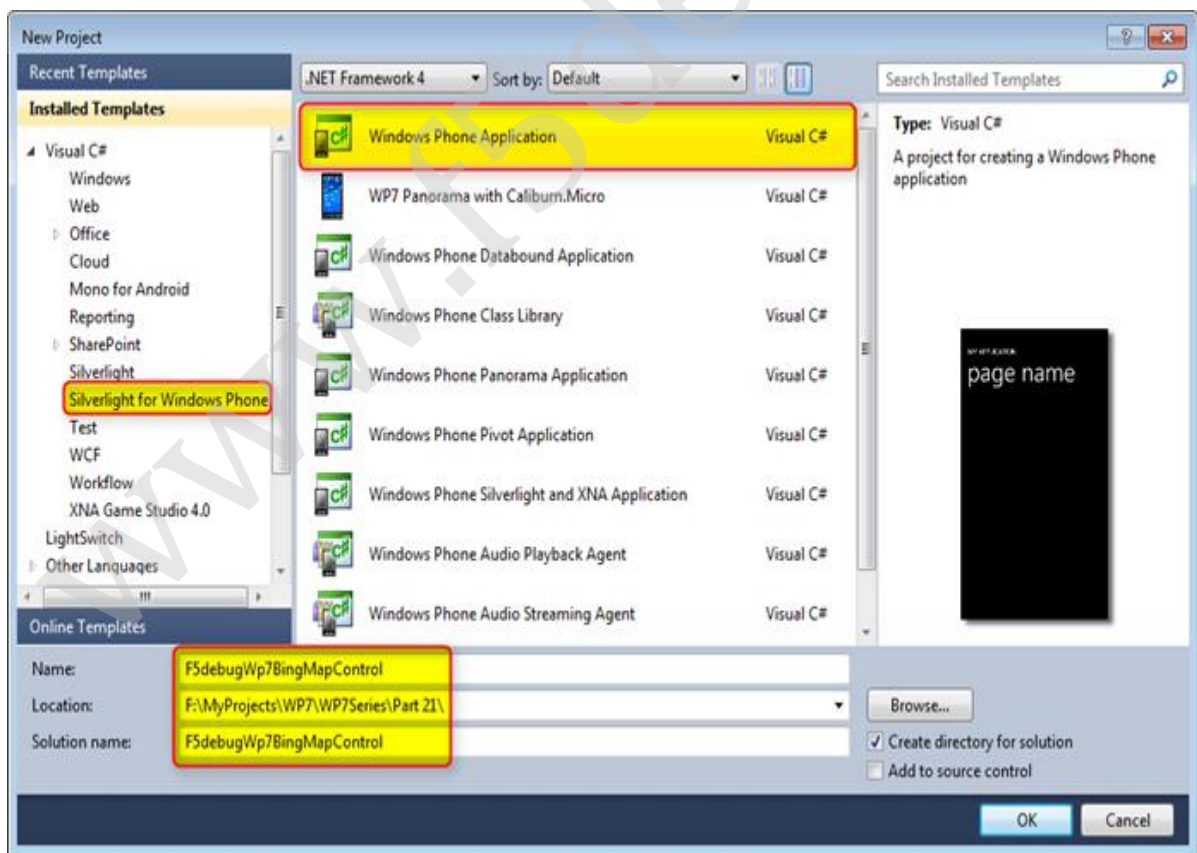
## Introduction:

In this article we are going to see the most useful control on the Windows Phone 7 development, yes the BING Map Control. The BING Map Silverlight control for Windows Phone 7 development combines the power of Silverlight and Bing maps to get the best mapping experience for the developers and the real users to create location based application.

BING Map control in Windows Phone 7 application development are derived from the name spaces Microsoft.Phone.Controls and Microsoft.Phone.Controls.Maps. Developers need to register for the BING Map developers account to get the keys which are used in developing the application with the Bing Maps. Here in this article we will see the step by step process on how to make use of the BING Maps control and get the most enriching map application.

## Steps:

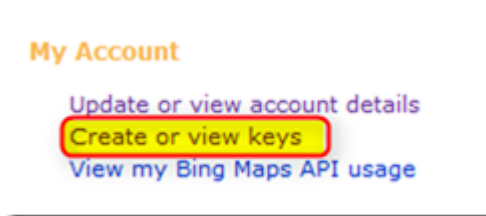
Open Visual Studio 2010 and create a new Silverlight for Windows Phone 7 application with a valid project name as shown in the screen below.



Note – To get the BING Map Keys first we need to register to the BING Map account using the below URL

<http://www.bingmapsportal.com/>

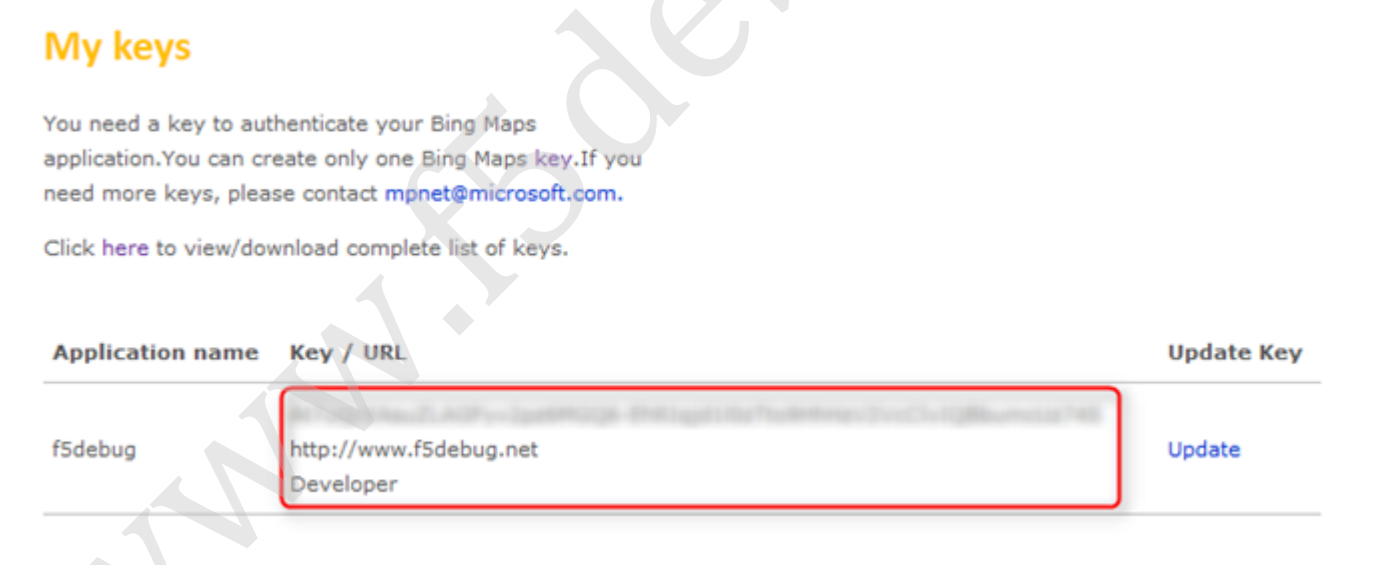
Once we register with our valid Windows Live ID we need to login again and go to the section Create or View Keys under the My Account category



Now we will be requested to provide the application details like below

- Application name: a valid application name
- Application URL: a valid application URL to access
- Application type: a valid application type to be selected from the list

Once we provided the details we will be provided with the BING Maps key as shown in the screen below.



Now we need to go back to the Visual Studio 2010 F5debugWp7BingMapControl project and start designing the application by dragging and dropping the BING Map control from the tool box and some buttons as shown in the screen below.

#### **XAML Code:**

```
<phone:PhoneApplicationPage
x:Class="F5debugWp7BingMapControl.MainPage"
xmlns="<a
href="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:phone="http://schemas.microsoft.com/winfx/2006/xaml/presentation">
```

```

xmlns:x="<a
href="http://schemas.microsoft.com/winfx/2006/xaml" href="http://schemas.microsoft.com/winfx/2006/xaml">http://schemas.microsoft.com/winfx/2006/xaml"</a>
xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
xmlns:d="<a
href="http://schemas.microsoft.com/expression/blend/2008" href="http://schemas.microsoft.com/expression/blend/2008">http://schemas.microsoft.com/expression/blend/2008"</a>
xmlns:mc="<a href="http://schemas.openxmlformats.org/markup-compatibility/2006" href="http://schemas.openxmlformats.org/markup-compatibility/2006">http://schemas.openxmlformats.org/markup-compatibility/2006"</a>
mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="696"
FontFamily="{StaticResource PhoneFontFamilyNormal}"
FontSize="{StaticResource PhoneFontSizeNormal}"
Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
shell:SystemTray.IsVisible="True" xmlns:my="clr-
namespace:Microsoft.Phone.Controls.Maps;assembly=Microsoft.Phone.Controls.Maps" Loaded="PhoneApplicationPage_Loaded">

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="Transparent">
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition Height="*" />
</Grid.RowDefinitions>

<!--TitlePanel contains the name of the application and page title-->
<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
<TextBlock x:Name="ApplicationTitle" Text="F5DEBUG WP7 TUTORIALS"
Style="{StaticResource PhoneTextNormalStyle}"/>
<TextBlock x:Name="PageTitle" Text="Bing Maps" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
</StackPanel>

<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
<my:Map Height="523" CredentialsProvider="At7uQrXAsuZLAGFyv2pz6MGQ6-
EhRIqjd1l0zTto9HhHzV2VcClvIQBbumcUz74S" HorizontalAlignment="Left"
Margin="6,6,0,0" Name="map1" VerticalAlignment="Top" Width="444" />
</Grid>
</Grid>

<!--Sample code showing usage of ApplicationBar-->
<phone:PhoneApplicationPage.ApplicationBar>
<shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
<shell:ApplicationBarIconButton IconUri="/Images/appbar.share.rest.png"
Text="Road View" Click="ApplicationBarIconButton_Click"/>
<shell:ApplicationBarIconButton
IconUri="/Images/appbar.feature.video.rest.png" Text="Aerial View"
Click="ApplicationBarIconButton_Click_1"/>
<shell:ApplicationBarIconButton IconUri="/Images/appbar.back.rest.png"
Text="Zoom In" Click="ApplicationBarIconButton_Click_2"/>
<shell:ApplicationBarIconButton IconUri="/Images/appbar.next.rest.png"
Text="Zoom Out" Click="ApplicationBarIconButton_Click_3"/>
</shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar>

</phone:PhoneApplicationPage>

```



Now we need to use the below using directive to start using the map properties to the code behind. So add the below code first to the using list at the top

### **C# Code:**

```
using Microsoft.Phone.Controls.Maps;
```

Now we need to add below code to the Road View and Aerial View button click events as shown in the code below.

### **C# Code:**

```
private void ApplicationBarIconButton_Click(object sender, EventArgs e)
{
    map1.Mode = new RoadMode();
}

private void ApplicationBarIconButton_Click_1(object sender, EventArgs e)
{
    map1.Mode = new AerialMode();
}
```

```
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Controls.Maps;

namespace F5debugWp7BingMapControl
{
    public partial class MainPage : PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }

        private void ApplicationBarIconButton_Click(object sender, EventArgs e)
        {
            map1.Mode = new RoadMode();
        }

        private void ApplicationBarIconButton_Click_1(object sender, EventArgs e)
        {
            map1.Mode = new AerialMode();
        }

        private void ApplicationBarIconButton_Click_2(object sender, EventArgs e)
        {
        }
    }
}
```

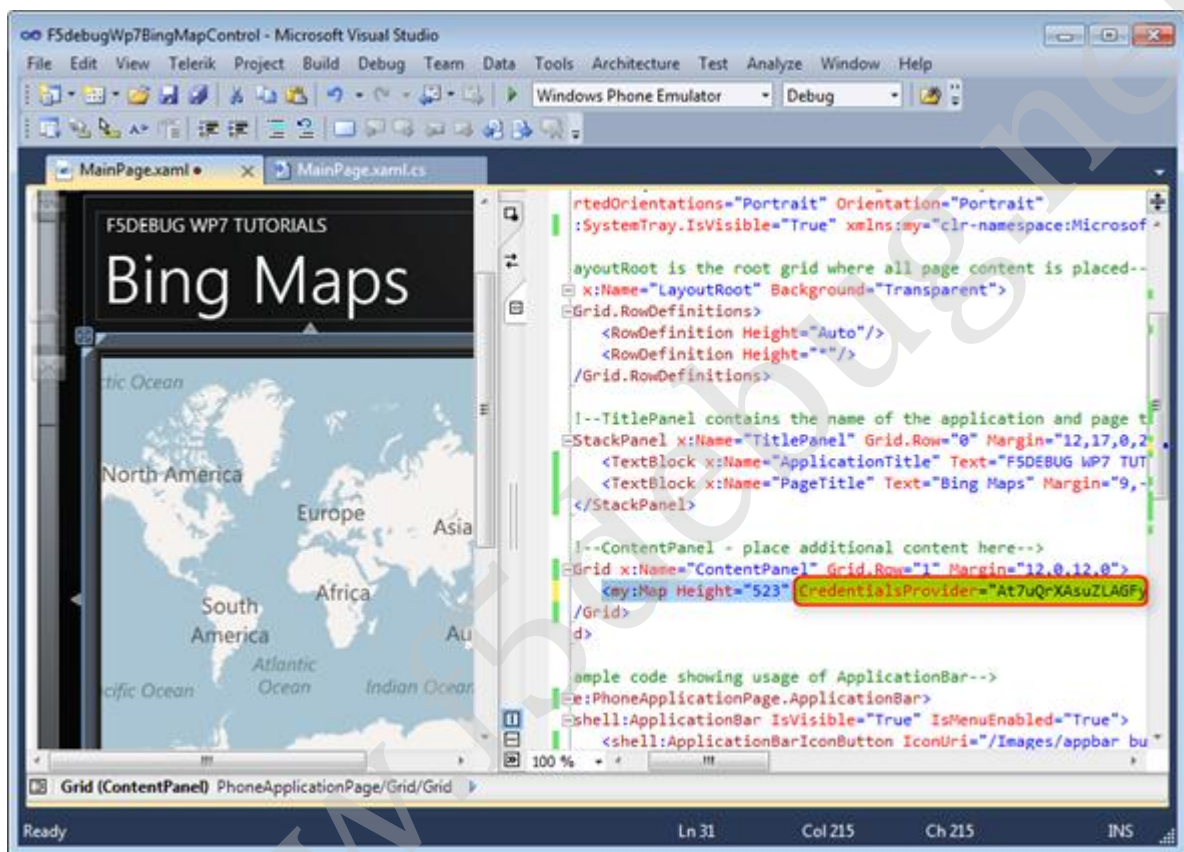
Now build and execute the application to check the first 2 button is working as expected. Press F5 to start building the project and we can see the application loaded to the Windows Phone 7 Emulator as shown in the screen below.



We can see a message Invalid Credentials: Sign up for a developer account at: <http://www.microsoft.com/maps/developers> as shown in the screen above. To remove this message we need to make use of the key which we got after registering with the Bing Maps website. So go to the XAML Code and add the key to the BING Maps section as shown in the screen below.

### XAML Code:

```
<my:Map Height="523" CredentialsProvider="Insert the KEY here"
HorizontalAlignment="Left" Margin="6,6,0,0" Name="map1"
VerticalAlignment="Top" Width="444" />
```



Now run the application again by pressing the F5 key and we can see the error message is removed and the application in Aerial View and Road View as shown in the screens below.

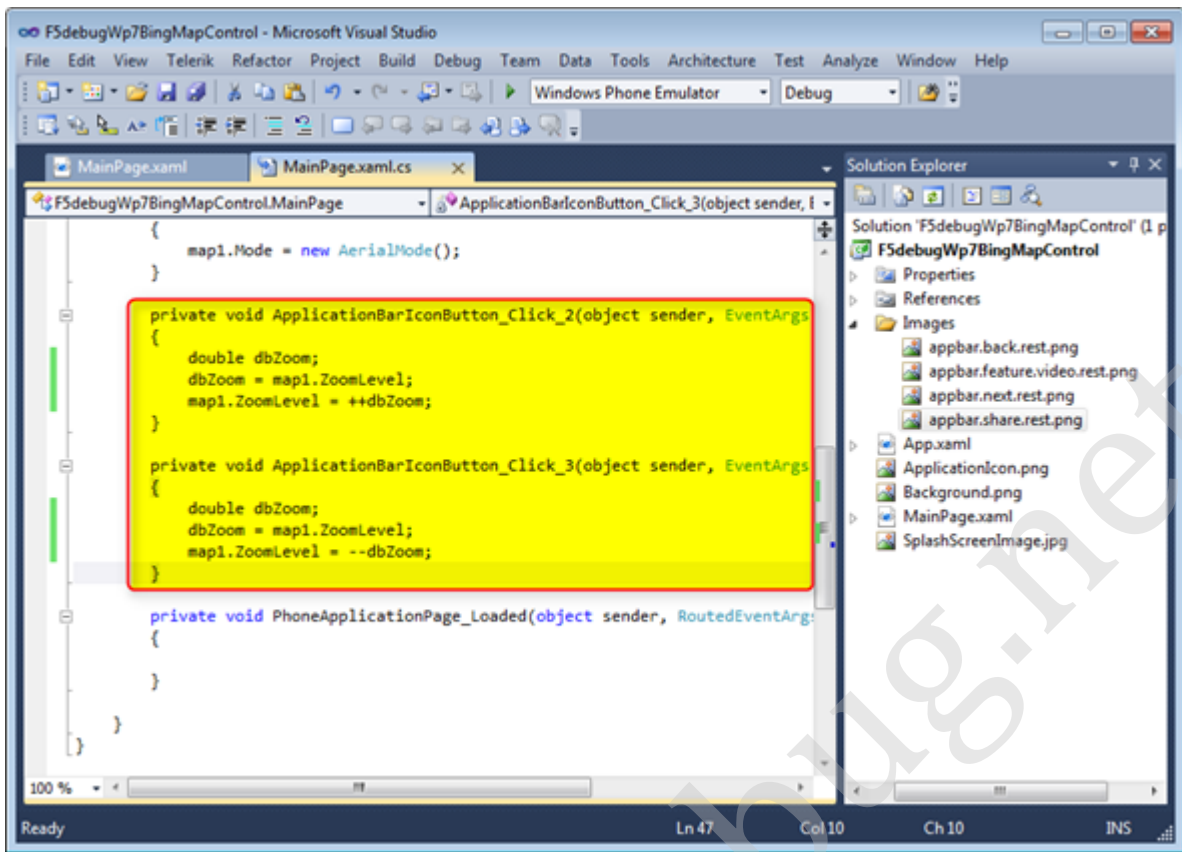


Now we will write our code to do the zooming for the maps. Write the below code in the respective code behind to get the Zoom in and Zoom out as shown in the code below.

**C# Code:**

```
private void ApplicationBarIconButton_Click_2(object sender, EventArgs e)
{
    double dbZoom;
    dbZoom = map1.ZoomLevel;
    map1.ZoomLevel = ++dbZoom;
}

private void ApplicationBarIconButton_Click_3(object sender, EventArgs e)
{
    double dbZoom;
    dbZoom = map1.ZoomLevel;
    map1.ZoomLevel = --dbZoom;
}
```



Now build and execute the project and we can see the Zoom in and Zoom out of the maps effectively as shown in the screens below.

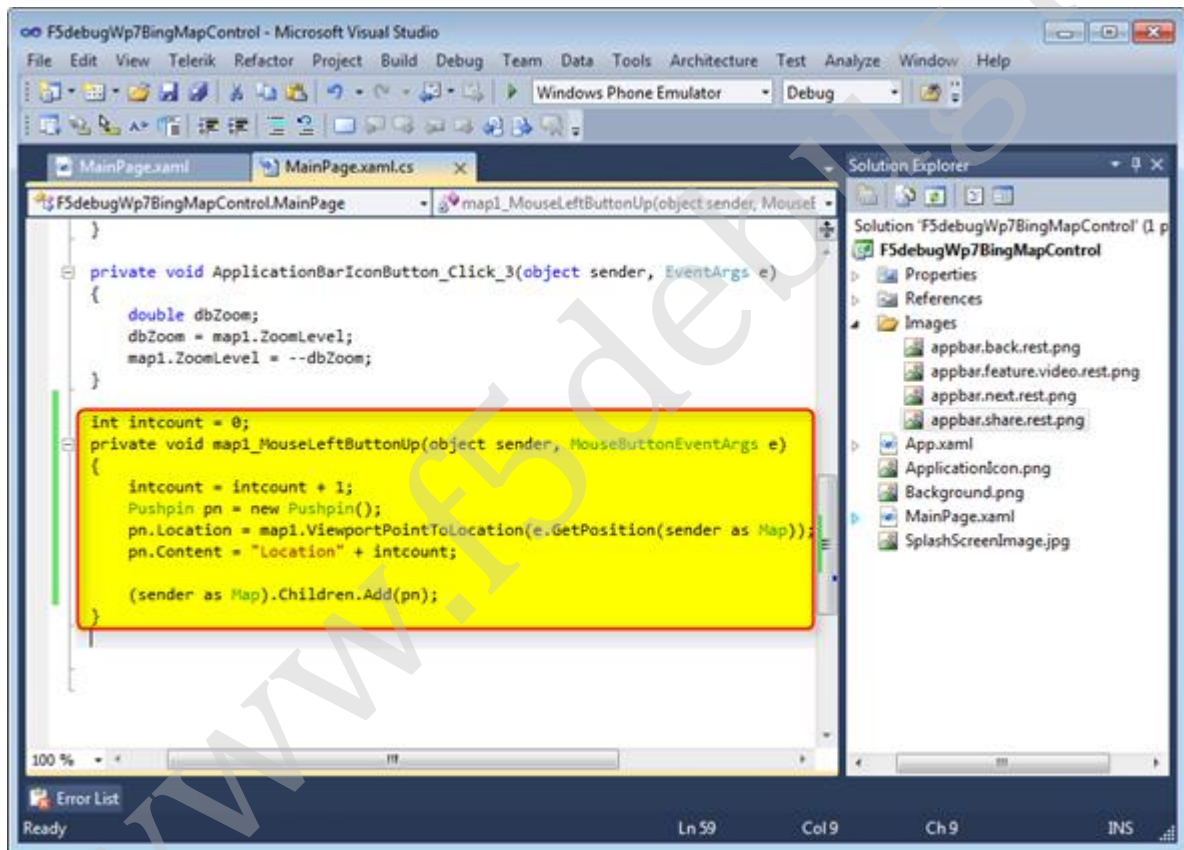


Now we got the expected output as shown in the above screens, now let us do a add on task of pinning the location normally we need to do while travelling using a map direction. To start coding the PINNING option first add the map1\_MouseLeftButtonUp event and write the below code which will do the location pinning as shown in the screen below.

### **C# Code:**

```
int intcount = 0;
private void map1_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    intcount = intcount + 1;
    Pushpin pn = new Pushpin();
    pn.Location = map1.ViewportPointToLocation(e.GetPosition(sender as Map));
    pn.Content = "Location" + intcount;

    (sender as Map).Children.Add(pn);
}
```



Now we are done with our code, let us build and execute the application by pressing F5 and we can see the expected output. Just use the Mouse Up to click on the location shown in the Windows Phone 7 Emulator to pin the location as shown in the screens below.



## Conclusion:

So in this article we have seen how to use the BING Map control effectively with the Windows Phone 7 development to build a rich user experience.

Thanks for reading my article. If you like my blog and if you are interested in getting the latest updates on new articles, kindly follow me through one of these options.

