Bing Maps Silverlight Control in Windows Phone 7 – TileSource and mobile tileset

During the last Mix'10, Microsoft unveiled the Windows Phone Developer tools which will be for sure, used in various project. Among them we will find mapping application with Bing Maps.

Through this article, I won't talk about the pros or the cons of Windows Phone 7 platform as I think there are more accurate people that can make the complete analysis.

Here, I would like to focus on the use where I can integrate in an easy way, mapping technology and more specially how I can integrate Bing Maps Silverlight control and make it works on Windows Phone.

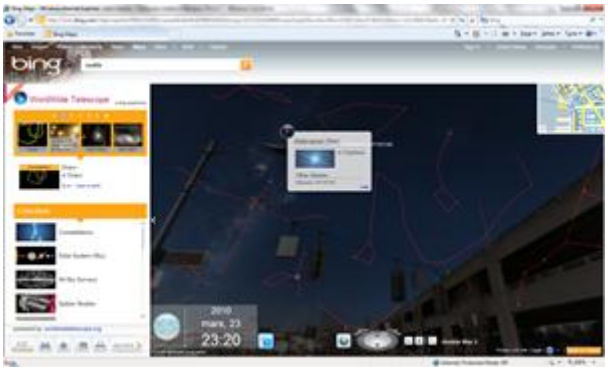
Small reminder about Bing Maps Silverlight Control

Bing Maps Silverlight Control is an official control published by Microsoft during 2009 even if we know that it was in developing process since a long time.

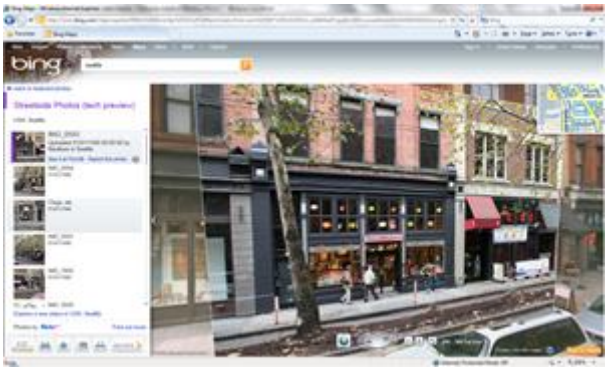
This control gives the possibility to use every features of Silverlight 3 and in the same time, it uses DeepZoom technology to get a really smooth animation on zooming or panning manipulation.

To discover the possibility, you can directly go to the iSDK website that can help to start or discover a specific feature:
http://www.microsoft.com/maps/isdk/silverlight/

The best website that uses this control is Bing Maps portal available here, with some really cool new features like the last one, integration of Worldwide Telescope experience:



Or the streetside photo which consists in integrating FlickR pictures directly on top of Streetside view included in Bing Maps:



Discovering Windows Phone 7 platform and tools

The new features and way that we need to work with Windows Phone 7 platform can be numerous but in fact if we take a look as a developer, we see that Microsoft take benefit from its technological background by using Silverlight 4 technology or XNA framework (v4).

The fact is that developers can developed now on Windows Phone 7 without having to learn a whole new environment. It's clearlya good thing and it helps company to get people ready to go on this platform.

Developer tools and ressources are available here:
http://developer.windowsphone.com/Default.aspx

They are directly integrated in VS2010 RC and other 2010 express version so developer are used to work with them.

Here are some good links to start or to get information:

- Official developer blog :
  http://windowsteamblog.com/blogs/wpdev/default.aspx
- MSDN resources :
  http://msdn.microsoft.com/en-us/library/ff402535(VS.92).aspx
- Here is an extract from the book written by Charles Petzold :
  http://download.microsoft.com/download/7/C/8/7C820C6F-C205-4ECF-B9F3-1505DD13F9BF/ProgWinPhonePreview.pdf
  Here is its full description :
  http://blogs.msdn.com/microsoft_press/archive/2010/03/15/free-ebook-programming-windows-phone-7-series-draft-preview.aspx

The mobile tile set and mapping images

Microsoft, through its mobile client shipped within Windows Phone 7 samples showed through events and that leaked somehow on the web use a specific tileset for mobile.

Again, I need to be clear on this point : there is no way this tileset can be used on commercial application as far as I know, it's just described for the sample.

This tileset exposes data through Microsoft's ECN (Edge caching network) like every other tileset. Here is a sample uri:

http://ecn.t7.tiles.virtualearth.net/tiles/cmd/mobileTile?g=330&base=h&a=21&device=mobile&dpi=d1

The available parameters in this uri are:

- tX : X number of the tile server (0 to 7)
- base=X : X mapstyle prefix (a=aerial, h=hybrid, r=road)
- a=QXYZ : QXYZ quadkey corresponding to the specific tile (X,Y and zoom)
- device=mobile : specifies thatwe use mobile tileset
- dpi=XX : map resolution set in hex value (« D1 » or « 40 » for example), i twill adapt the labels size on the map

We can get these kind of tiles for example :

Integrating mapping data and tileset

First, we need to disable the base tile layer, andto do this a simple code has to be added as a child of the MapControl:

```
<m:Map.Mode>
    <mc:MercatorMode />
</m:Map.Mode>
```

To add the mobile tile set, we need to create a specific class that will implement TileSource class and override its GetUri() methods :

```
public class BingMapsMobileTileSource : TileSource
{
    #region Properties

    public string MapDpiResolution { get; set; }
    public string MapStyle { get; set; }

    #endregion

    public BingMapsMobileTileSource()
        : base("http://ecn.t{2}.tiles.virtualearth.net/tiles/cmd/
mobileTile?g=330&base={1}&a={0}&device=mobile&dpi={3}")
    {

    }

    #region TileSource specific methods

    public override Uri GetUri(int x, int y, int zoomLevel)
    {
        string quadKey = QuadKeyHelper.TileXYToQuadKey(x, y, zoomLevel);
        return new Uri(
            string.Format(
                this.UriFormat,
                quadKey,
                this.MapStyle,
                (x % 7),
                this.MapDpiResolution
            )
        );
    }

    #endregion
}
```

We can add some basic property to ease the use of uri parameters like map resolution or map style information.

Then we have to register the namespaces that will be used on declarative part (XAML code), here are the namespace definition part:

```
xmlns:m="clr-
namespace:Microsoft.Maps.MapControl;assembly=Microsoft.Maps.MapControl"
xmlns:mc="clr-
namespace:Microsoft.Maps.MapControl.Core;assembly=Microsoft.Maps.MapControl
"
xmlns:local="clr-namespace:SLBingMobile.Utilities.CustomTileSource"
```

Here is the declarative markup that permits to integrate our Bing Maps control with all options and layer information.

```
<Grid x:Name="ContentGrid" Grid.Row="1">
    <m:Map NavigationVisibility="Collapsed">
        <m:Map.Mode>
            <mc:MercatorMode />
        </m:Map.Mode>
        <m:MapTileLayer>
            <m:MapTileLayer.TileSources>
                <local:BingMapsMobileTileSource
                    MapDpiResolution="d1"
                    MapStyle="r">
                </local:BingMapsMobileTileSource>
            </m:MapTileLayer.TileSources>
        </m:MapTileLayer>
    </m:Map>
</Grid>
```

Here is a result when we launch the application:

Conclusion on this integration

Here is a simple way to integrate Bing Maps Silverlight Control as a part of a Windows Phone Application. We only add some tile on top of the control.

But the fact is that it shows how it is possible to do this in a simple way and how Microsoft gives us (developers☺) the possibility to code and run our applications on many types of device.

You can download the sample source code here:
http://www.boonaert.net/sources/SLBingmobile.zip

Through future articles, I will use this control in some basic case so we can find an interest for this kind of sample.