Creating an app which shares your position on FaceBook

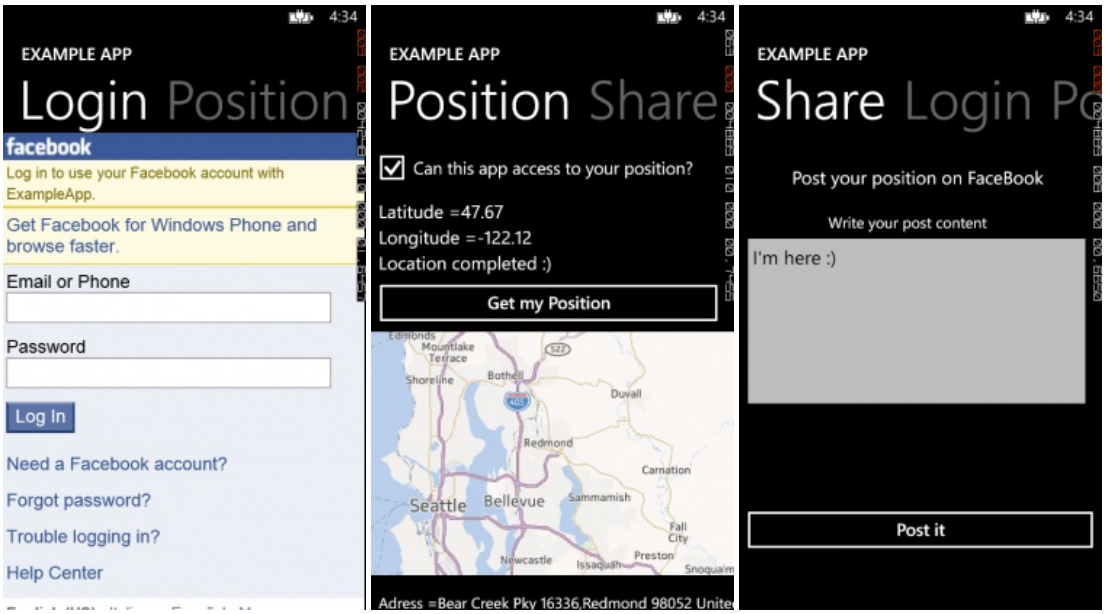
This article show show to create a real application that uses many of the topics described in the Wiki. It includes Facebook integration, Single-Shot Location, the new Nokia <Map /> control and other topics.

Introduction

The idea behind this app is very simple: get your position, traslate it into an address, show it into the map and share it with your friends on Facebook. To achieve this in an app we need a few tools:

- **Facebook Integration** : to make the Login and to Post on user's wall
- **Single-Shot Location**: to get user's actual position, and show it into the map
- **Nokia <Map /> Control**: to show user's position into the map

The App will look like this:



Create the UI

The structure of the UI is very simple: we need a *pivot* for making the user navigate into the 3 main tabs.

Each of the 3 tabs will have a specific function:

- The first one will let the user to log-in on facebook and authorize our app to write on his wall
- The second one will get user location, and will show it into the map
- The last one will create the message and post it

So, let's write the code

```
//This will create the pivot object

<phone:Pivot x:Name="appPivot" HorizontalAlignment="Left" Height="768" Title="EXAMPLE APP" VerticalAlignment="Top" Width="480" Grid.

//This is the first PivotItem. It will show a WebBrowser, which will open the Login Page
                <phone:PivotItem x:Name="login" CacheMode="{x:Null}" Header="Login" Margin="0,0,0,-8">
                <phone:WebBrowser x:Name="mWebBrowser" HorizontalAlignment="Left"
                          Margin="0,0,0,0" Grid.Row="1"
                          VerticalAlignment="Top"
                          Height="643" Width="478"
                          IsScriptEnabled="True"
                          Navigated="WebBrowser_Navigated"/>

            </phone:PivotItem>

//This is the second PivotItem, with all information about user's position
//Because the Pivot Item can have only one content we use Canvas to include multiple content inside it
            <phone:PivotItem x:Name="position" CacheMode="{x:Null}" Header="Position" Margin="0,10,-2,-6">
                <Canvas HorizontalAlignment="Left" Height="631" VerticalAlignment="Top" Width="480">
                <CheckBox
                   Content="Can this app access to your position?"
                          HorizontalAlignment="Center"
                          VerticalAlignment="Top"
                          Margin="0,0,0,0" Grid.Row="1"
                          Checked="AllowTracking"
                          Unchecked="DisallowTracking"/>
                    <TextBlock x:Name="Latitude" HorizontalAlignment="Center"
```


(C) Copyright Nokia 2013. All rights reserved.

```
                    TextWrapping="Wrap" Text="Latitude ="
                    VerticalAlignment="Top" FontSize="24" Canvas.Top="77" Canvas.Left="10"/>
            <TextBlock x:Name="Longitude" HorizontalAlignment="Center"
                    TextWrapping="Wrap" Text="Longitude ="
                    VerticalAlignment="Top" FontSize="24" Canvas.Top="111" Canvas.Left="10"/>
            <TextBlock x:Name="Status" HorizontalAlignment="Center"
                    TextWrapping="Wrap" Text="Status ="
                    VerticalAlignment="Top" FontSize="24" Canvas.Top="145" Canvas.Left="10"/>
            <Button x:Name="TrackButton" Content="Get my Position" Canvas.Top="179" Width="470" Click="getPosition"/>
            <Grid x:Name="ContentPanel" Grid.Row="1" Canvas.Top="253" Height="322" Width="480">
                <maps:Map x:Name="MyMap" ZoomLevel="10"/>
            </Grid>
            <TextBlock x:Name="StreetPos" HorizontalAlignment="Center"
                    TextWrapping="Wrap" Text="Adress ="
                    VerticalAlignment="Top" FontSize="20" Canvas.Top="597" Canvas.Left="10"/>

        </Canvas>


    </phone:PivotItem>

//The Last pivot show a textbox in which the user could insert his message content and a button to post it
    <phone:PivotItem x:Name="share" CacheMode="{x:Null}" Header="Share" Margin="0,28,-2,-6" Height="613" VerticalAlignment="
        <Canvas HorizontalAlignment="Left" Height="613" Grid.RowSpan="2" VerticalAlignment="Top" Width="480">

            <TextBlock HorizontalAlignment="Center" TextWrapping="Wrap" Text="Post your position on FaceBook" Width="340" Fo
            <TextBox x:Name="postContent" Height="242" TextWrapping="Wrap" Text="I'm here :)" Width="470" Canvas.Top="100"/>
            <Button Content="Post it" Canvas.Top="461" Width="480" Click="post"/>
            <TextBlock TextWrapping="Wrap" Text="Write your post content" Canvas.Left="118" Canvas.Top="76" TextAlignment="C


        </Canvas>

    </phone:PivotItem>
</phone:Pivot>
```

Let's go ahead and write the AI of the App

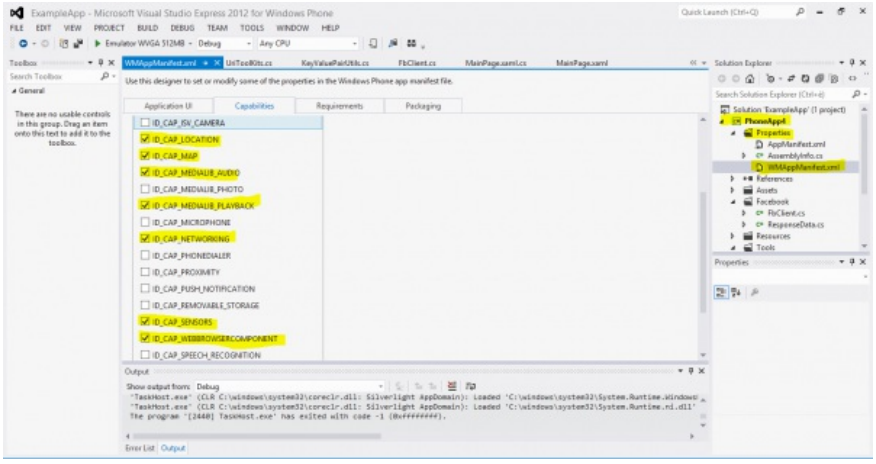
Add the required capabilites and using statement

To make the app properly works we need to add many capabilities to it. For doing it follow these instructions:

1. Create a new **Windows Phone App** and then select **Windows Phone 8.0** as platform you want to target
2. Go to **Solution Explorer**, expand the **Properties** folder, and then double-click **WMAppManifest.xml**
3. Click on the **Capabilities** tab of the manifest designer, select the check box next to

- **ID_CAP_LOCATION**
- **ID_CAP_MAP**
- **ID_CAP_SENSORS**
- **ID_CAP_WEBBROWSERCONTENT**

Final result will be like the image:

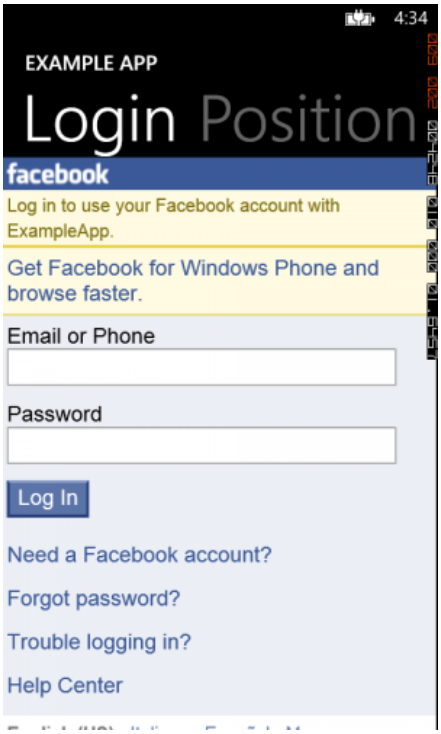


Then move to the **MainPage.cs** and add the following using statements

```
using PhoneApp4.Resources;
using System.IO.IsolatedStorage;
using Windows.Devices.Geolocation;
using System.Device.Location; // Contains the GeoCoordinate class.


using Tools;
using Facebook;
using Microsoft.Phone.Maps.Services;
using System.IO;
using System.Runtime.Serialization.Json;
using System.Text;
```

First Page: Login

(C) Copyright Nokia 2013. All rights reserved.



The aim of the first page is to let the user login on FaceBook and authorize our app. This operation will return an Access Token which will be used to post content

Before of writing the code we have to do some operation:

- You have to register your app on Facebook Developer. So go to this link https://developers.facebook.com/apps and click on **Create New App** (you have to sign up first)
- Enter the **Name App** and click on **Continue**
- Enter **Display Name** and **NameSpace** and then click on **Save Changes**
- The above operation will give you the **AppID** and **App Secret** keys which you should save as you will have to use them in your code later on.

Let's now focus on the code. In the UI we will create a PivotItem which will only have as content add a WebBrowser item. This will have **IsScriptEnabled** property on **true**, and the event handler for the event **Navigate**(in this example: WebBrowser_Navigated)

```
<phone:Pivot x:Name="appPivot" HorizontalAlignment="Left" Height="768" Title="EXAMPLE APP" VerticalAlignment="Top" Width="480" Grid.

//This is the first PivotItem. It will show a WebBrowser, which will open the Login Page
                <phone:PivotItem x:Name="login" CacheMode="{x:Null}" Header="Login" Margin="0,0,0,-8">
                <phone:WebBrowser x:Name="mWebBrowser" HorizontalAlignment="Left"
                          Margin="0,0,0,0" Grid.Row="1"
                          VerticalAlignment="Top"
                          Height="643" Width="478"
                          IsScriptEnabled="True"
                          Navigated="WebBrowser_Navigated"/>

            </phone:PivotItem>
 //WRITE OTHER PIVOT ITEM

</phone:Pivot>
```

Before anything, you have to add 4 files to our project. These would be **FbClient.cs**, **ResponseData.cs**, **UriToolKits.cs** and **KeyValuePairUtils.cs**. A short explanation of how to do this is provided in Integrate Facebook to Your Windows Phone Application#Add_Library

Moving to the **MainPage.cs** page, we have to clean user's cookies and to set the *source* property for `WebBrowser`. This action must be done into the **MainPage Constructor**

```
public MainPage()
        {
           InitializeComponent();

            // Clear Cookie to remove current logged in user data
            mWebBrowser.ClearCookiesAsync();

            // Go to Login url
            mWebBrowser.Source = new Uri(FbClient.Instance.GetLoginUrl());
        }
```

Now write `WebBrowserNavigated` event handler and `AccessTokenDownloadCompleted` method. Both these are well explained in the article Integrate Facebook to Your Windows Phone Application

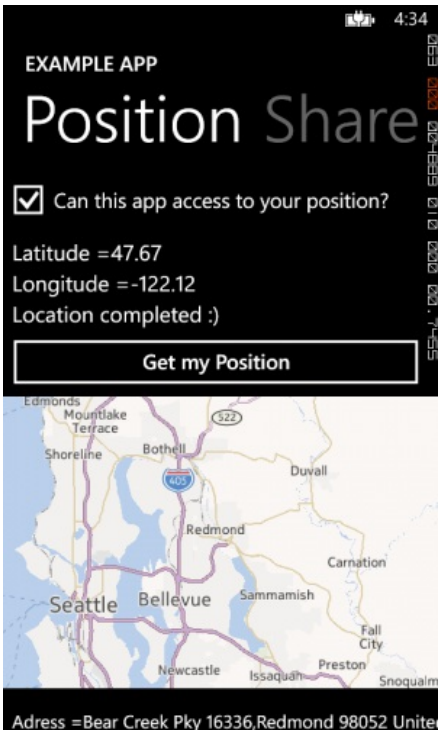
```
        private void WebBrowser_Navigated(object sender, NavigationEventArgs e)
        {
            String uri = e.Uri.ToString();
            if (uri.StartsWith("https://www.facebook.com/connect/login_success.html") || uri.StartsWith("http://www.facebook.com/con
            {
                // Remove junk text added by facebook from url
                if (uri.EndsWith("#_=_"))
                    uri = uri.Substring(0, uri.Length - 4);
```

 (C) Copyright Nokia 2013. All rights reserved.

```
            String queryString = e.Uri.Query.ToString();

            // Acquire the code from Query String
            IEnumerable<KeyValuePair<string, string>> pairs = UriToolKits.ParseQueryString(queryString);
            string code = KeyValuePairUtils.GetValue(pairs, "code");

            // Get access_token from code using Asynchronous HTTP Request
            WebClient client = new WebClient();
            client.DownloadStringCompleted += new DownloadStringCompletedEventHandler(AccessTokenDownloadCompleted);
            client.DownloadStringAsync(new Uri(FbClient.Instance.GetAccessTokenRequestUrl(code)));
        }
    }


    void AccessTokenDownloadCompleted(object sender, DownloadStringCompletedEventArgs e)
    {
        string data = e.Result;
        data = "?" + data;
        // Acquire access_token and expires timestamp
        IEnumerable<KeyValuePair<string, string>> pairs = UriToolKits.ParseQueryString(data);
        string accessToken = KeyValuePairUtils.GetValue(pairs, "access_token");
        string expires = KeyValuePairUtils.GetValue(pairs, "expires");

        // Save access_token
        FbClient.Instance.AccessToken = accessToken;

        // Back to MainPage
        appPivot.SelectedItem = position;
    }
```

Now the first page is complete. Move to the second

Get user's position and show it into the map



The second tab will be focused on this action

Since we need to use the <Map /> element, we need to declare it at the beginning of the .xaml file. So Add this declaration in the top of the code

```
xmlns:maps="clr-namespace:Microsoft.Phone.Maps.Controls;assembly=Microsoft.Phone.Maps"
```

Here is the code for the UI of the *Position* `PivotItem`. Since `PivotItem` could contains only one content,you need to use Canvas as a container for everything.

You will need as content:

- **CheckButton** For register user agreement in using his position
- **3 TextBlock** For visualizing coordinates of the user's position and the status of location
- **Map** For showing the user's position
- **Another TextBlock** For write user's actual address

```
<phone:PivotItem x:Name="position" CacheMode="{x:Null}" Header="Position" Margin="0,10,-2,-6">
               <Canvas HorizontalAlignment="Left" Height="631" VerticalAlignment="Top" Width="480">
               <CheckBox
                  Content="Can this app access to your position?"
                         HorizontalAlignment="Center"
                         VerticalAlignment="Top"
                         Margin="0,0,0,0" Grid.Row="1"
```


(C) Copyright Nokia 2013. All rights reserved.

```
                        Checked="AllowTracking"
                        Unchecked="DisallowTracking"/>
                    <TextBlock x:Name="Latitude" HorizontalAlignment="Center"
                               TextWrapping="Wrap" Text="Latitude ="
                               VerticalAlignment="Top" FontSize="24" Canvas.Top="77" Canvas.Left="10"/>
                    <TextBlock x:Name="Longitude" HorizontalAlignment="Center"
                               TextWrapping="Wrap" Text="Longitude ="
                               VerticalAlignment="Top" FontSize="24" Canvas.Top="111" Canvas.Left="10"/>
                    <TextBlock x:Name="Status" HorizontalAlignment="Center"
                               TextWrapping="Wrap" Text="Status ="
                               VerticalAlignment="Top" FontSize="24" Canvas.Top="145" Canvas.Left="10"/>
                    <Button x:Name="TrackButton" Content="Get my Position" Canvas.Top="179" Width="470" Click="getPosition"/>
                    <Grid x:Name="ContentPanel" Grid.Row="1" Canvas.Top="253" Height="322" Width="480">
                        <maps:Map x:Name="MyMap" ZoomLevel="10"/>
                    </Grid>
                    <TextBlock x:Name="StreetPos" HorizontalAlignment="Center"
                               TextWrapping="Wrap" Text="Adress ="
                               VerticalAlignment="Top" FontSize="20" Canvas.Top="597" Canvas.Left="10"/>

                </Canvas>


            </phone:PivotItem>
```

The C# code will include:

2 Method for registering user's agreement or disagreement is tracking his Location. User will give it by checking/unchecking the `CheckBox` To save his decision you have to use `IsolatedStorageSettings`

```
 private void AllowTracking(object sender, RoutedEventArgs e)
        {
            IsolatedStorageSettings.ApplicationSettings["LocationConsent"] = true;
        }

        private void DisallowTracking(object sender, RoutedEventArgs e)
        {
            IsolatedStorageSettings.ApplicationSettings["LocationConsent"] = false;
        }
```

Then we have to write the event handler `getPosition`, which will be called when the user will press on the button *Get my position*. This is an **async** method, so it will requires an **await** status inside it.

If the users didn't agree in using his position, this method will return immediately. Otherwise it will get user position by the method `geolocator.GetGeopositionAsync()`. To set user's position on the map, set **Center** property of the <Map /> object, equal to the user's Latitude and Longitude.

Finally, to convert the coordinate into an address, use `GeoCoordinate` class. You will need to write an event handler for the event **QueryCompleted** of the `GeoCoordinate` class

```
private async void getPosition(object sender, RoutedEventArgs e)
        {
            if (!IsolatedStorageSettings.ApplicationSettings.Contains("LocationConsent"))
            {
                Status.Text = "Status = I can't access your position :(";
                return;
            }
            else if ((bool)IsolatedStorageSettings.ApplicationSettings["LocationConsent"] != true)
            {
                Status.Text = "Status = I can't access your position :(";
                return;
            }

            Geolocator geolocator = new Geolocator();
    geolocator.DesiredAccuracyInMeters = 5;

    try
    {
        Geoposition geoposition = await geolocator.GetGeopositionAsync(
             maximumAge: TimeSpan.FromMinutes(5),
             timeout: TimeSpan.FromSeconds(10)
            );

//With this 2 lines of code, the app is able to write on a Text Label the Latitude and the Longitude, given by Geoposition
        Latitude.Text = "Latitude ="+geoposition.Coordinate.Latitude.ToString("0.00");
        Longitude.Text = "Longitude ="+ geoposition.Coordinate.Longitude.ToString("0.00");
        Status.Text = "Location completed :)";
        MyMap.Center = new GeoCoordinate(geoposition.Coordinate.Latitude, geoposition.Coordinate.Longitude);
        ReverseGeocodeQuery myAdress= new ReverseGeocodeQuery();
        myAdress.GeoCoordinate=new GeoCoordinate(geoposition.Coordinate.Latitude, geoposition.Coordinate.Longitude);
        myAdress.QueryCompleted+=myAdress_QueryCompleted;
        myAdress.QueryAsync();

    }
//If an error is catch 2 are the main causes: the first is that you forgot to includ ID_CAP_LOCATION in your app manifest.
//The second is that the user doesn't turned on the Location Services
    catch (Exception ex)
    {
        if ((uint)ex.HResult == 0x80004004)
        {
            Status.Text = "Location is disabled in phone settings";
        }
        else
        {
             Status.Text = "Some error happened during the acquisition of the location";
        }
    }
}
```

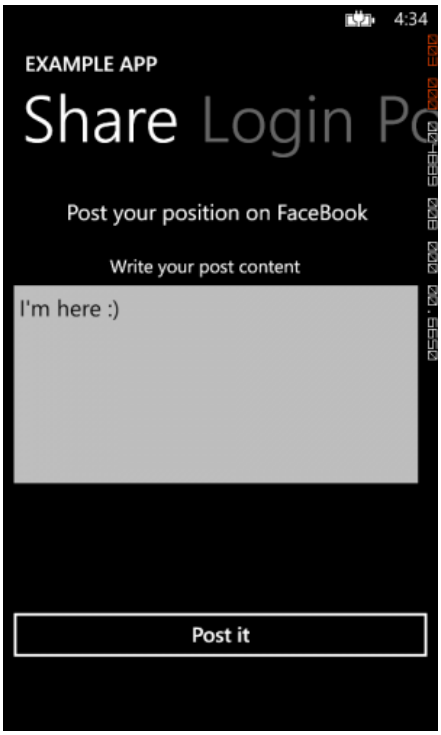
So let's write the `QueryCompleted` event handler. This method will extract the needed information from the output of the method **QueryAsync**, saving it into a String, that will be wrote in the UI, into the apposite `TextBlock`.

 (C) Copyright Nokia 2013. All rights reserved.

```
void myAdress_QueryCompleted(object sender, QueryCompletedEventArgs<IList<MapLocation>> e)
{

   foreach (var item in e.Result)
    {

        adress+=item.Information.Address.Street+" "+item.Information.Address.HouseNumber+","+item.Information.Address.City;
       adress+=" "+item.Information.Address.PostalCode+" "+(item.Information.Address.Country);

    }
   StreetPos.Text = "Adress =" + adress;


}
```

Also this page is complete! Let's move to the last

Post message on FaceBook



In the last `PivotItem`, you will have a `TextBox` in which the user will insert his text, and a button for posting it into his wall.

So here's the xaml code:

```
<phone:PivotItem x:Name="share" CacheMode="{x:Null}" Header="Share" Margin="0,28,-2,-6" Height="613" VerticalAlignment="Top">
                <Canvas HorizontalAlignment="Left" Height="613" Grid.RowSpan="2" VerticalAlignment="Top" Width="480">

                    <TextBlock HorizontalAlignment="Center" TextWrapping="Wrap" Text="Post your position on FaceBook" Width="340" Fo
                    <TextBox x:Name="postContent" Height="242" TextWrapping="Wrap" Text="I'm here :)" Width="470" Canvas.Top="100"/>
                    <Button Content="Post it" Canvas.Top="461" Width="480" Click="post"/>
                    <TextBlock TextWrapping="Wrap" Text="Write your post content" Canvas.Left="118" Canvas.Top="76" TextAlignment="C
                </Canvas>
            </phone:PivotItem>
```

Moving to the C# part of the application you need to write an handler for the button, that will post the user's message on Facebook. For doing this, you will use the `PostOnWallCompleted` method of the class `FacebookClient`. We will give as input to this method the content of the message that the user has written, added of the user's address of the position in which he is.

```
private void post(object sender, RoutedEventArgs e)
{
    FbClient.Instance.PostMessageOnWall(postContent.Text+" "+adress, new UploadStringCompletedEventHandler(PostMessageOnWallComplete
}
```

For documentation about `PostMessageOnWall` and `UploadStringCompletedEventHandler`, i recommend you to the article Integrate Facebook to Your Windows Phone Application#Post_to_Wall

The app is now complete

Documentation

For more info about all the tools used in this app i recommend you to the articles of the topics interested in this app

- Get_Phone_Location_with_Windows_Phone_8

(C) Copyright Nokia 2013. All rights reserved.

- Asynchronous_Programming_For_Windows_Phone_8
- Integrate_Facebook_to_Your_Windows_Phone_Application
- Introduction_and_best_practices_for_IsolatedStorageSettings
- http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj207045(v=vs.105).aspx

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.