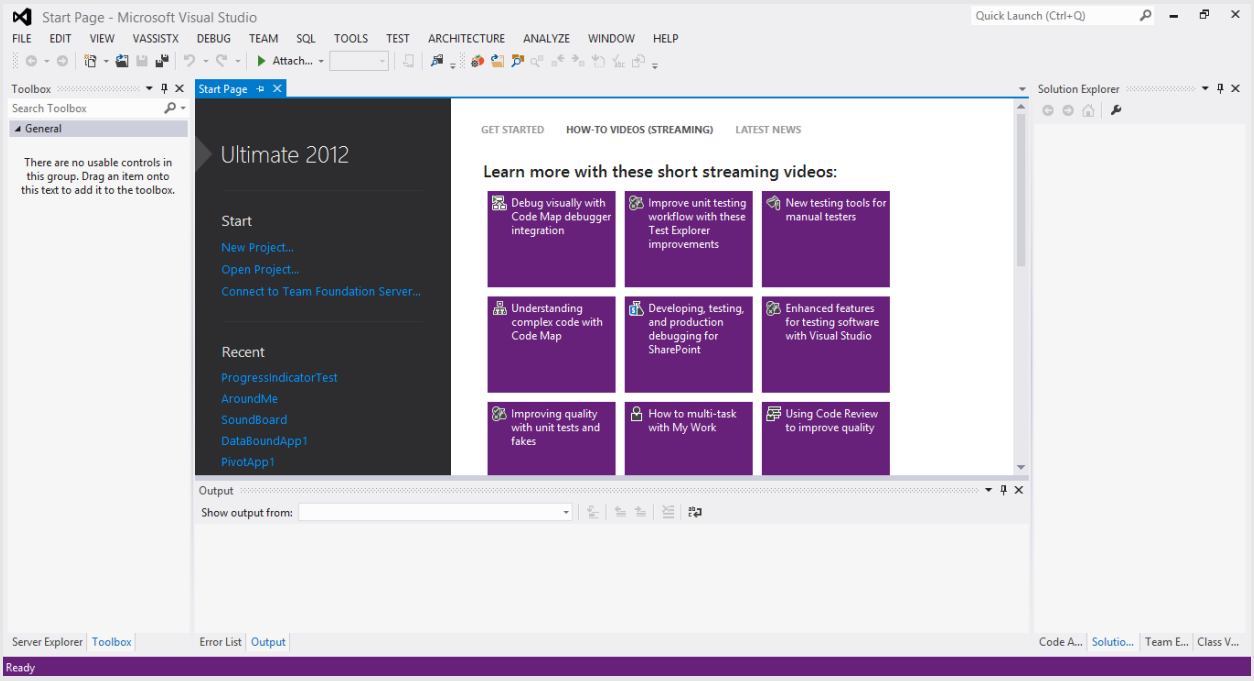
After we successfully installed the Windows Phone SDK, it's now we get ourselves rolling on the developing our first app for WP8.

In this tutorials, I would take you through step by step in understanding the in depth part of developing an app.

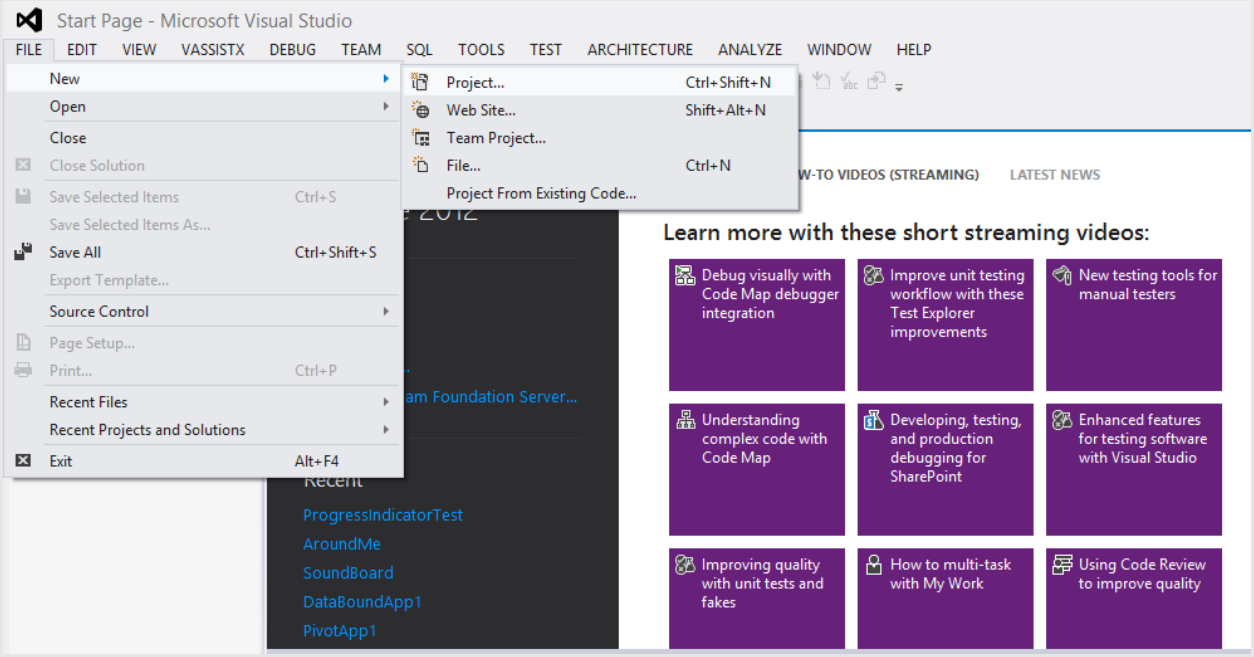
Start you visual studio and it takes you to the home screen.

When you open Visual Studio, you'll get yourself to the application's start page. There is a lot of content on this page, including development tips, recent news, and project related actions.



It is sometimes helpful to browse the content here to learn more about the platform, but for now just click on the “New Project…” link in the left sidebar.

Or navigate to the top bar, Click on ‘File’….Select ‘New’ and then Project!



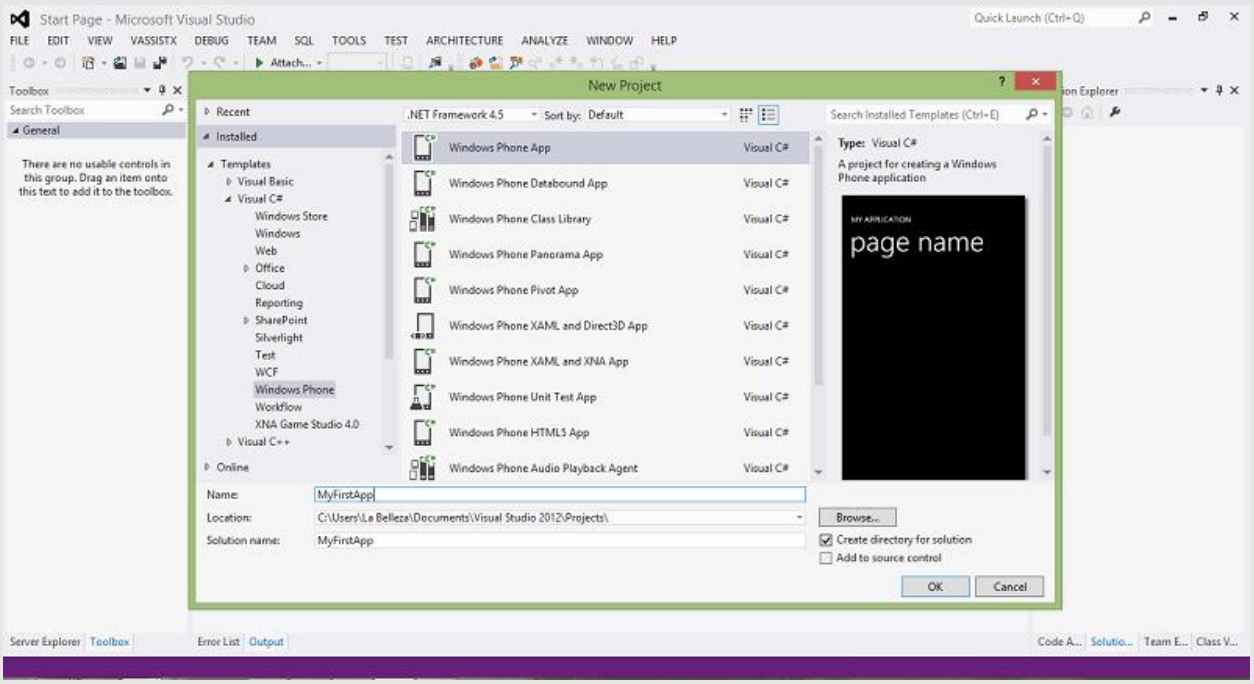
A dialog box will pop up that guides you through creating your new project.

On the ‘**Left Bar’,** click on ‘Templates’ and you will get the extended menu.

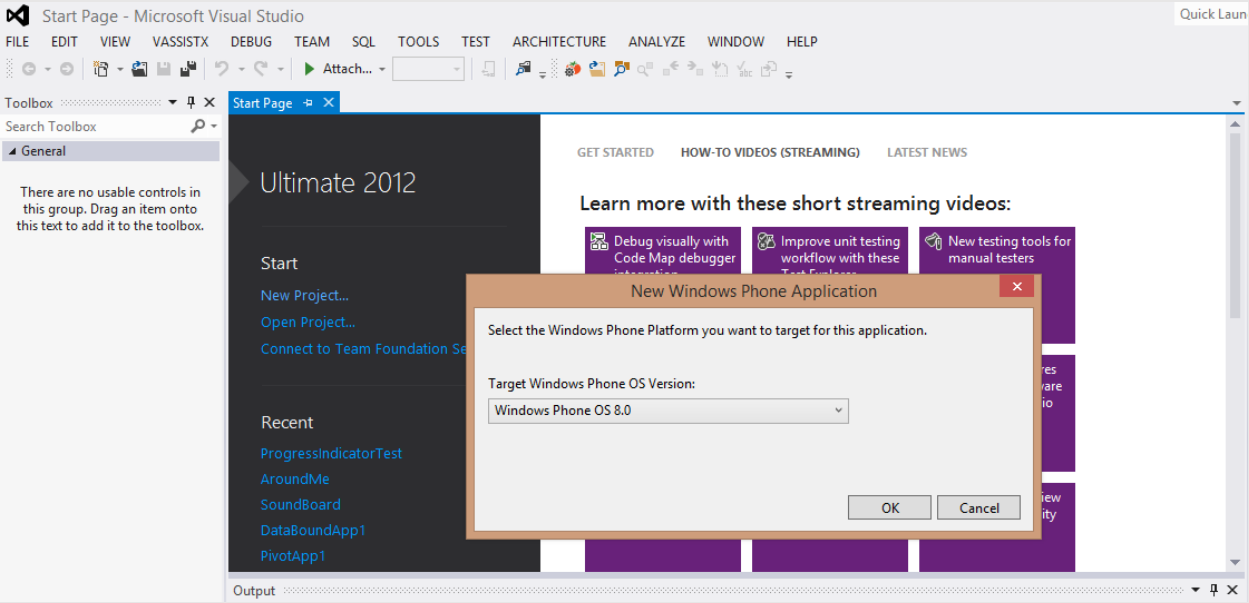
For now we will build our apps using the '**Visual c#',** select on '**Windows Phone Application'** and a list of the templates are listed for you on the middle panel.

Select the first one '**Windows Phone App'** and on the right panel you should see the '**phone template'** with your app having the 'Page Name'.

On the bottom panel of the dialog, give your application a name on the **'Name'** space, on the **'Location',** give the directory to your preference storage for your application and click **'OK'.**



From this dialog, you get another dialog on top of it if you had emended your Windows Phone SDK on a visuals Studio 2012 that options you to create app for Windows Phone 7, but for now select Windows Phone 8.0



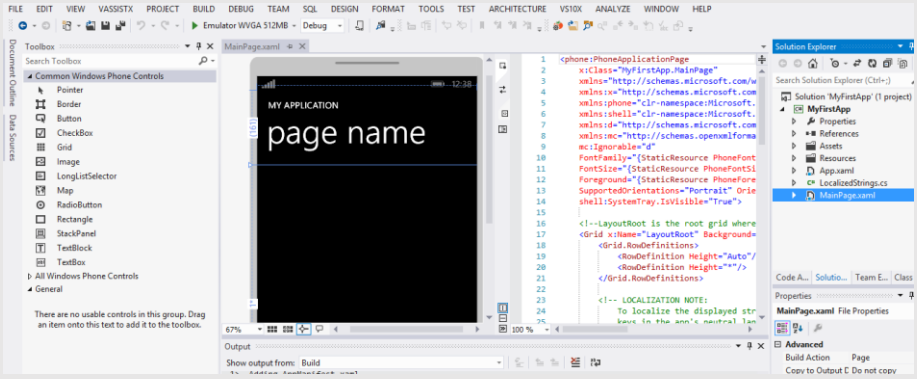
This will now take us to a new display where we can play around with our first application.

Before we do so, let us get a quick overview of the provided new project.

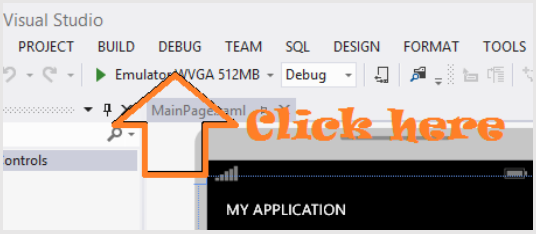
On the left bar, we have the **'toolbox'** that contains the **'Common Windows Phone Controls'** that we will use later on our application by drag and dropping them on our **'Display Layout'.**

The middle display of our project we have the **'Display Layout'** and the **'Xaml Page'** where one can edit the app from.

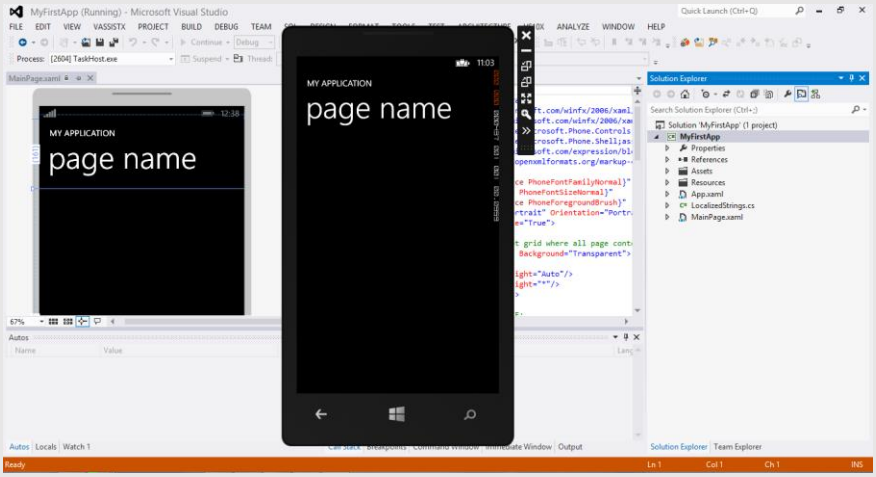
The right panel contains the '**Solution Window',** if it's now visible, click on the top bar **'View -> Solution Explorer'** or press **'Ctrl+Alt+L'.**



Now we can build our application by clicking on **'Build -> Build Solution'** or press **'Ctrl+Shift+B'.** If your app solution builds up with no error, now you can go ahead and test you application on the emulator by pressing **'f5'** or clicking on the default emulator selection.



The emulator takes some seconds to start and few more to deploy your application on the emulator before it starts to run.



Getting back to our project that you just created using the Silverlight for Windows Phone application template, you will realize that it contains a number of files and folders. I would like us to review them to get a better understanding of what goes into a Windows Phone application. On the right Panel of the visual studio, we have the '**Solution Explorer**'.

For any default Windows Phone Silverlight Application auto generated project have's the following files:

a) App.xaml and App.xaml.cs:

This files defines the application's entry point and the first page (screen) that gets loaded. It also contains styles that are part of a global resource dictionary that can be reference on the whole project.

b) MainPage.xaml and MainPage.xaml.cs:

The 'MainPage.xaml' contains the UI elements and their code behind it is on 'MainPage.xaml.cs'. You can get to this by pressing **'f7**' while at the .xaml page. MainPage.xaml is by default the application start page as defined in App.xaml.

c) References folder:

This folder contains a list of assemblies **'.NET DLLs'** that provide services and functionality for the application.

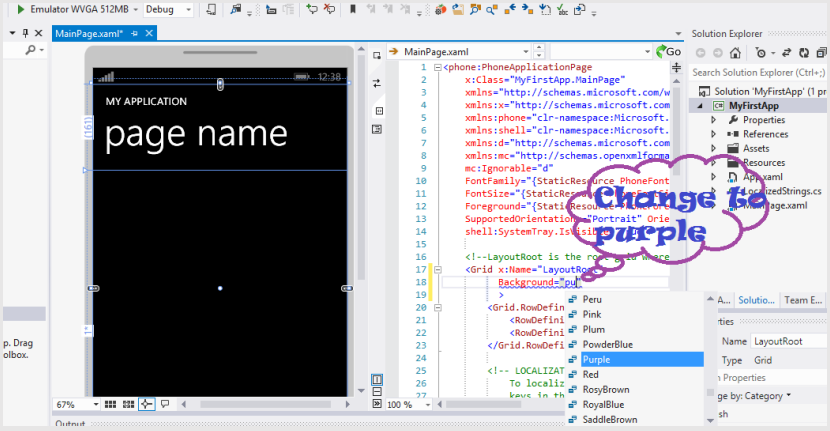
d) The properties folder:

This folder includes additional files which usually developer don't have to work with, as they are auto generated and managed by the IDE but one needs to do some change on app submission!

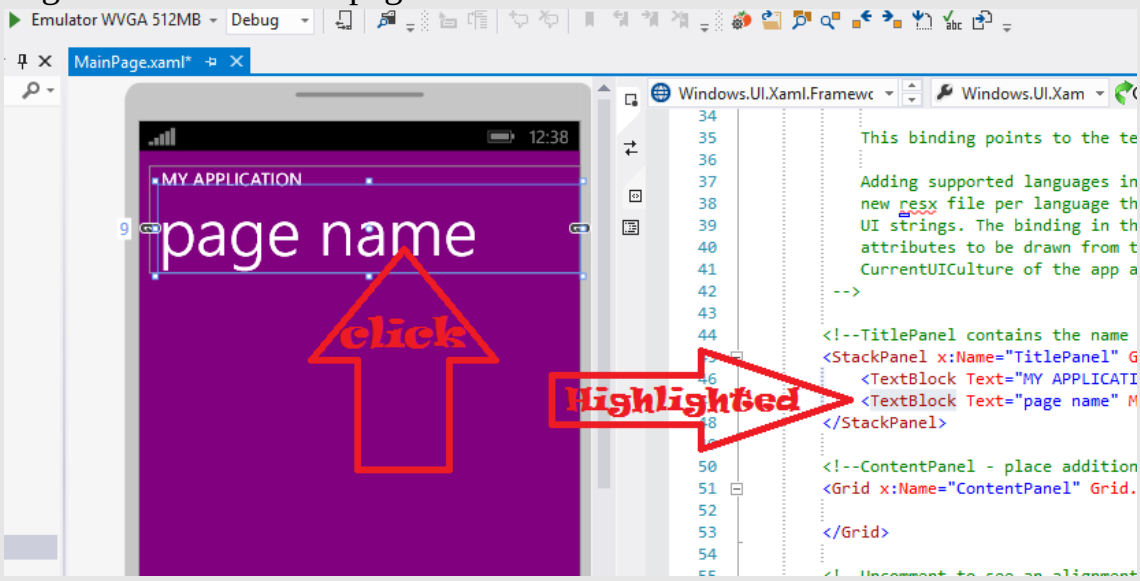
a. **AppManifest.xml:** This manifest file contains details on everything in the packaged WP Silverlight application, such as the different DLLs that are part of the application.
b. **AssemblyInfo.cs:** This file contains the metadata information about the assembly.
c. **WMAppManifest.xml:** this is a manifest file that includes specific metadata related to a Windows Phone application.

Playing around with the Application.

A. Changing the **Background color.**



a. Let's also change the app title
   a. Click on the text block having the context 'Page Name' and it will be Highlited on the .xaml page.



   b. Change the Text of your textblock to be **'Home Page'**
   c. Do the same by clicking on 'MY APPLICATION' textblock and change the content to be **'MyFirstApp'**

b. **Adding a TexBox**
   a. On the left panel, we have the tool Box, drag and drop a **'TextBox'** to the display layout of your application.
   b. On the .xaml page we have the properties of the textbox being displayed.

c. Let's edit the property "`Text="TextBox"`" and remove the text in between the tags to be like this "`Text=""`".
d. Add a property for the name of your texbox, '`Name="_myTextBox"`'

c. Adding a TextBlock

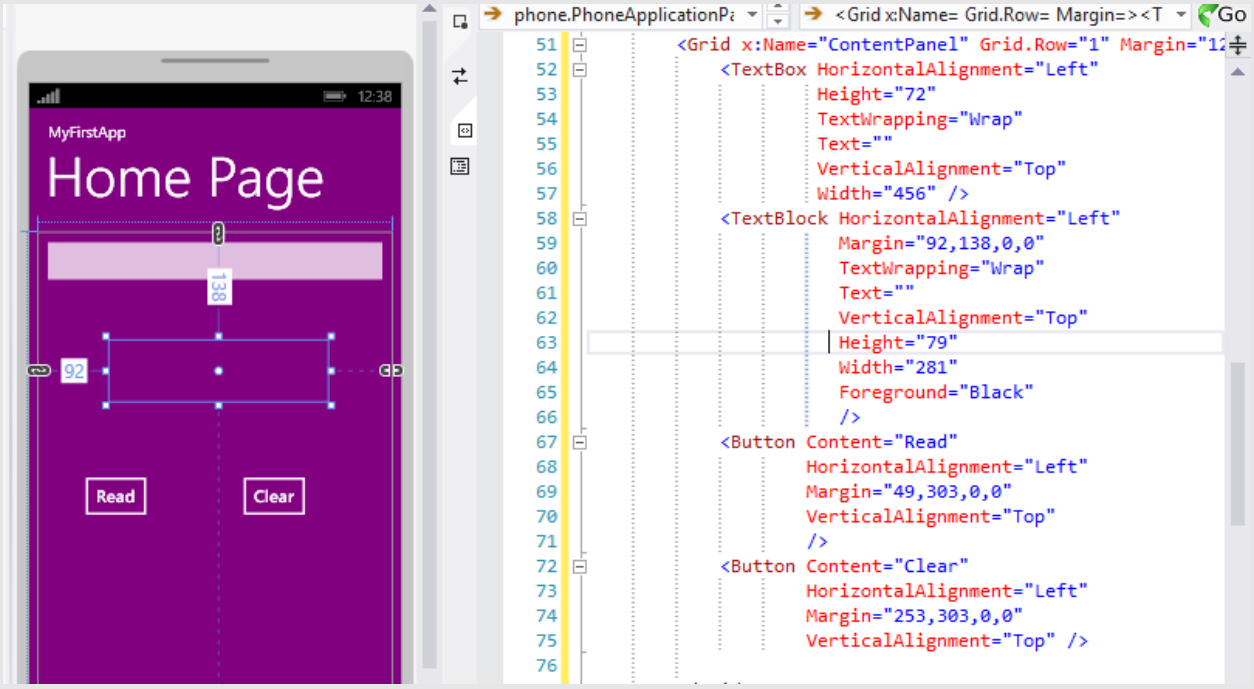
a. On the left panel, we have the tool Box, drag and drop a **'TextBlock'** to the display layout of your application.
b. Let's edit the property "`Text="TextBlock"`" and remove the text in between the tags to be like this "`Text=" "`".
c. Also lets add one more property for the text appearance by adding the property **'**`Foreground="Black"`**'**
d. Add a property for the name of your texbox, '`Name="_myTextBlock"`'

d. **Adding a button.**

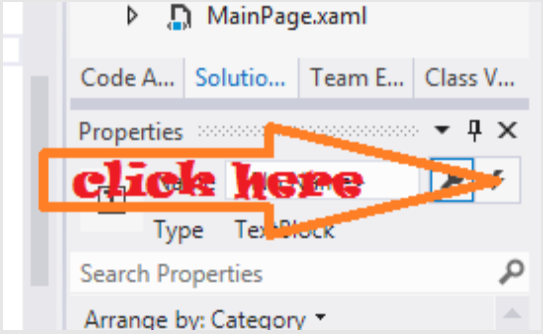
a. On the left panel, we have the tool Box, drag and drop a **'button'** to the display layout of your application.
b. On the .xaml page, the button properties are displayed for you to edit.
c. Edit the property "`Content="Button""`" and remove the text in between the tags to be like this "`Content="Read"`".

e. **Adding a button.**

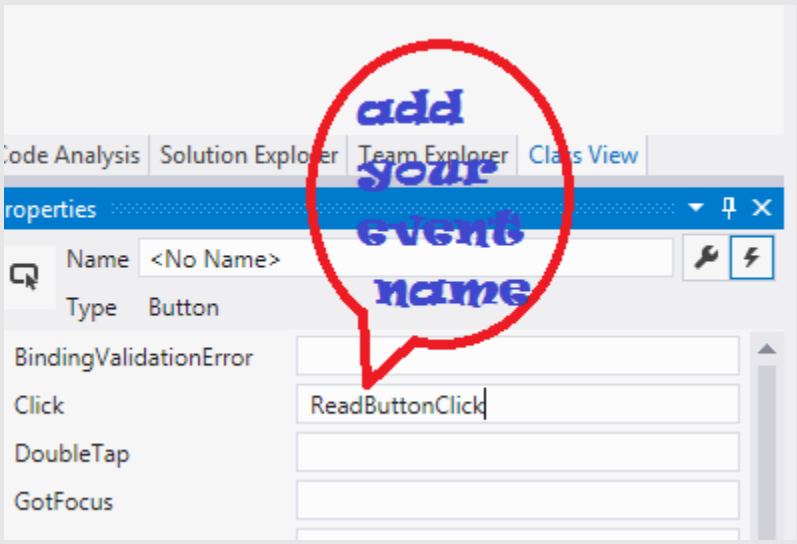
a. On the left panel, we have the tool Box, drag and drop a **'button'** to the display layout of your application.
b. On the .xaml page, the button properties are displayed for you to edit.
c. Edit the property "`Content="Button""`" and remove the text in between the tags to be like this "`Content="Clear"`".



f. Adding event
   a. Click on the button and on the Right Panel to the lower end, we have the **'property window'.**
   b. Click on the event icon.



   c. Add the click event name.



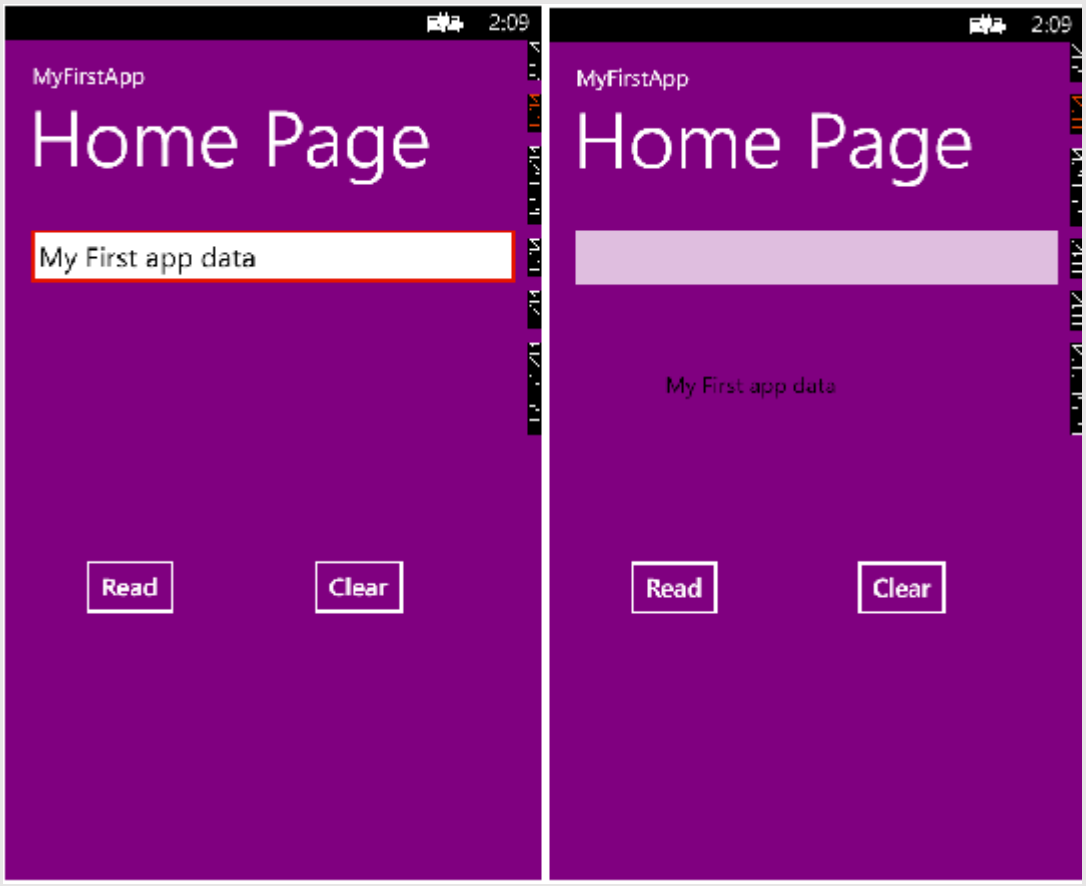
d. Press Enter and it will take you to the MainPage.xaml.cs page.
e. On the .cs page, add the following code to read the text from the textbox and display it on the textblock.

f. Add an event for the Button **'Clear'** to clear the text on the textBlock.

```
private void ReadButtonClick(object sender, RoutedEventArgs e)
{
    //Add an your event here
    //1. Get the text from the textBox and store it in a string.

    string text = _myTextBox.Text;
    //2. Display the text on the textBlock created

    _myTextBlock.Text = text;

    //3. Clear the textBox
    _myTextBox.Text = "";
}

private void ClearButtonClick(object sender, RoutedEventArgs e)
{
    //1. Clear the text on the textXlock
    _myTextBlock.Text = "";

}
```

g. Now let us run our application.

h. Press the **'PageDn'** button on your pc to use the pc keyboard.
i. I hope everything run up well on your emulator.



We shall continue from here on our next tutorials.
Thanks.

Kamiri Peterson M,
Game & Software Developer, Tutor & Mentor.
Google+: http://gplus.to/pbosoa
Blog: www.bhakitah.blogspot.com
Quick Links: www.kapes.yolasite.com
Nokia Store Links: Nokia Store
Windows Store Links: Windows Store