

Message Box, Application Bar and Page Navigation

Part 3



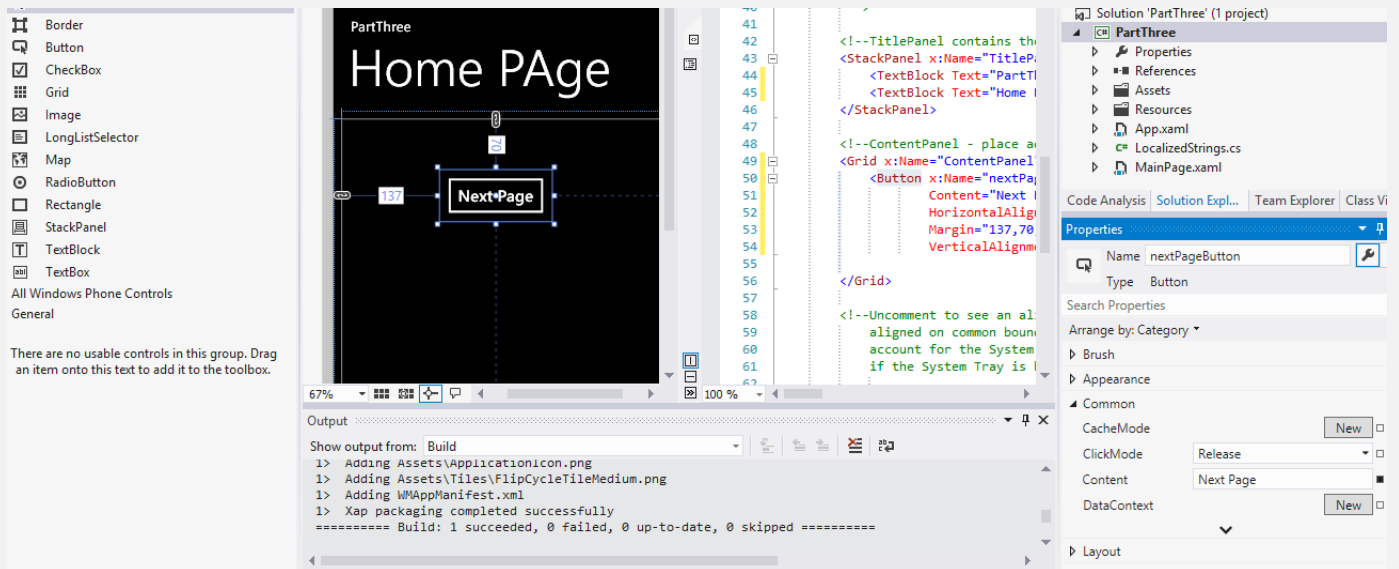
Welcome back to WP8 programming, in this tutorials, we will get into deep dive on how to be able to use the Message Box, Application Bar and Page Navigation together with Navigating data from one page to another.

A. Page Navigation:

Open the visual studio and create a new WP8 application 'PartThree'. Navigation on a WP8 app is enabled by the presence of the PhoneApplicationPage's class that has the property NavigationService. This service is the one used to navigate from one page to another. This NavigationService exposes the Navigate method which loads the specified page you want to navigate to.

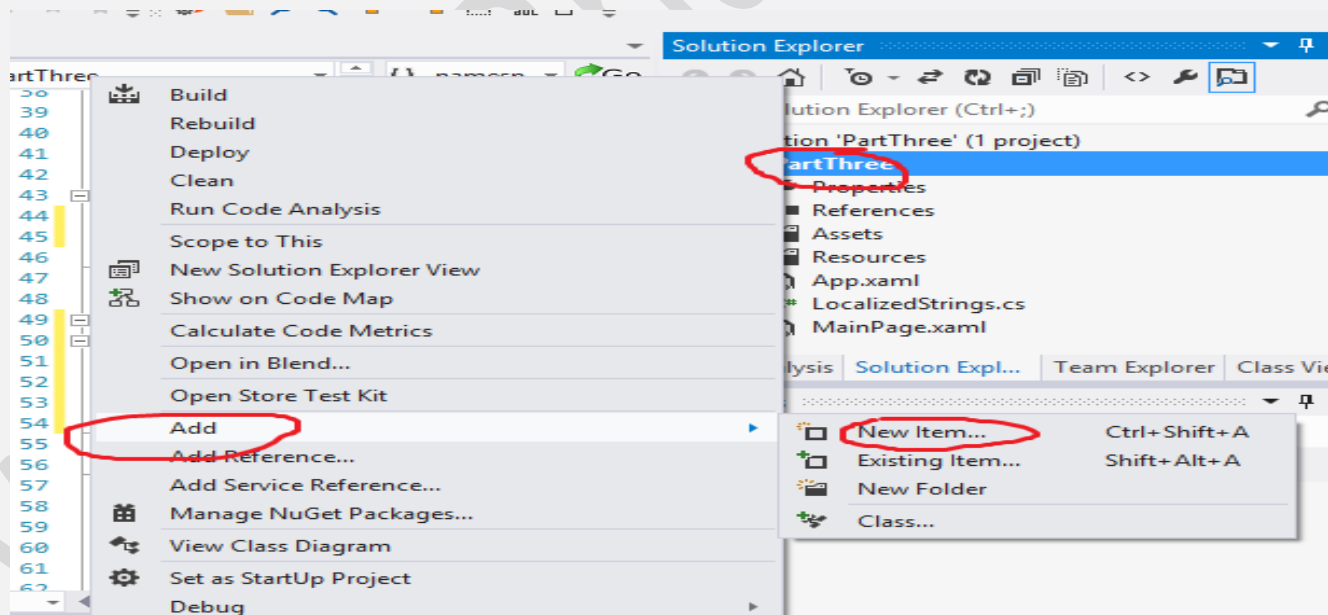
1. Adding a Button.

On the Display Layout, let's drag and drop a button from the ToolBox Window on the Left and rename its content as "Next Page" on the Property Window, also give the Button a name, preferably "nextPageButton"!

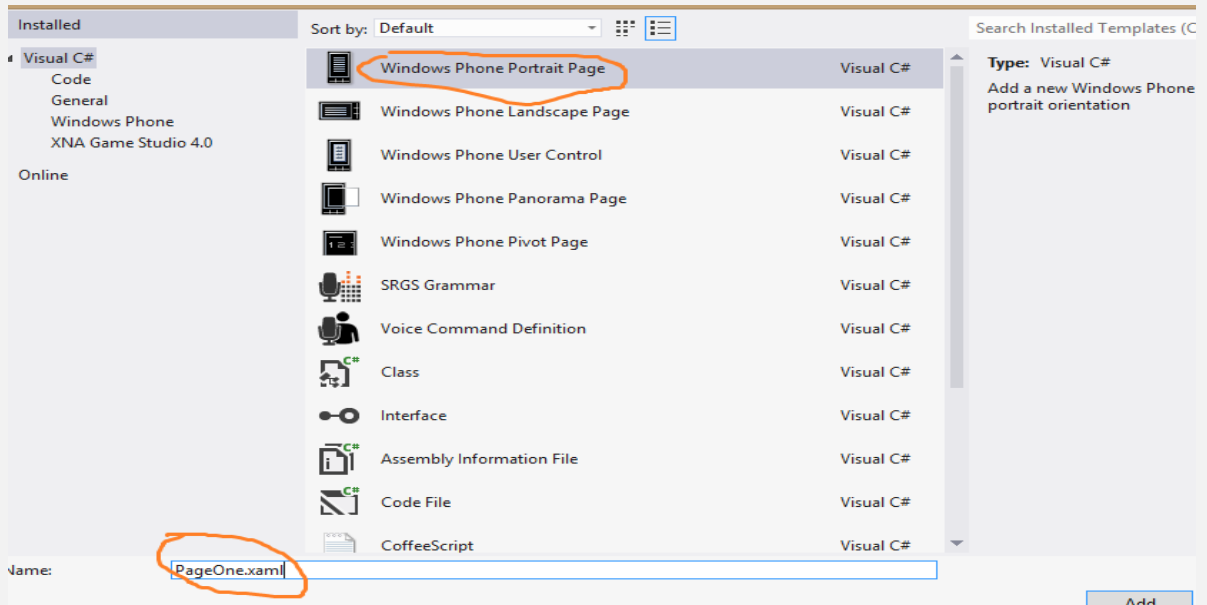


2. Creating a new Page

To have a new Windows Phone Application Page, right click on the project on the “Solutions window” and select Add -> New Item. Or Press ‘Ctrl+Shift+A’



Choose Windows Phone Portrait Page and rename it to PageOne.xaml.



3. Creating the event.

On the home page, double click the “nextPageButton” and it creates the event handler for you.

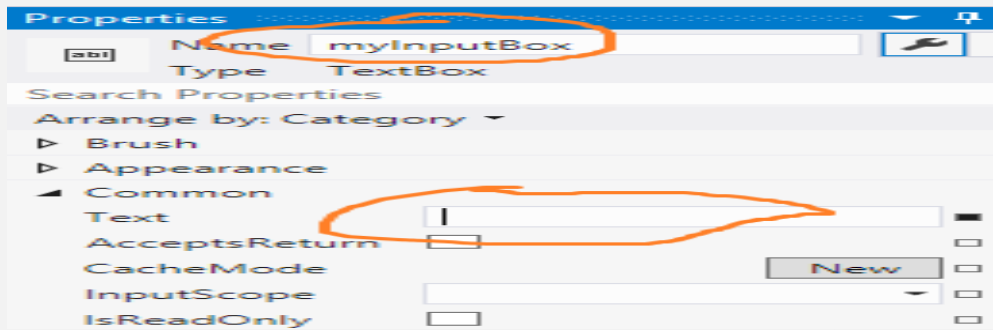
Add the following C# code for navigation.

```
private void nextPageButton_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new Uri("/PageOne.xaml", UriKind.Relative));
}
```

Press f5 to run your new app.

B. Page navigation with Data.

When developing a WP8 application, sometimes it requires one to navigate with data from one Windows Phone Application Page to the next. For it to be possible, you need to create a “**PhoneApplicationService.State dictionary**” that will temporary store your data. The storage persists until your app is deactivated. To receive the data on our PageOne.xamk, we will have to overied ‘**onNavigatedTo()**’ method. Add a ‘TextBox’ by drag and dropping from the ‘ToolBox’ window on your Display Layout Page on the Home Page and rename it as “myInputBox” and remove the text.



1. On the navigation event for your button, add the following code.

```
private void nextPageButton_Click(object sender, RoutedEventArgs e)
{
    //1. have a sting to temporary hold the data
    string myData = myInputBox.Text;
    //2. Have dictionary name that will enable you receive data.
    //this is more of a name tag for each and every piece of data
    //to be send and received on the navigated to page.
    NavigationService.Navigate(new Uri("/PageOne.xaml?data="+myData, UriKind.Relative));
}
```

2. On PageOne, drag and drop a “textBlock” on the “DisplayLayout” from the “ToolBox”. Rename the textblock as “myDsisplayTextBlock”. Add ‘onNaviagtedTo’ event.

```
protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
{
    base.OnNavigatedTo(e);
    //1. have a string to hold the data received temporary
    string recivedData = "";
    //2. get the data and store it on our buffer string
    if (NavigationContext.QueryString.TryGetValue("data", out recivedData)) ;
    //3. Display the data
    myDsisplayTextBlock.Text = recivedData;
}
```

3. Now run the app again!

C. MessageBox

Message Box in WP8 are used to display to the user a scenario where the information is displayed to the user, the message is disposable only when the user selects to ‘OK’ or when provided with a choice to accept or cancel. Lets re visit our sample application and use a message box to Display the data user inputs in the ‘textBox’.

1. Drag and drop an new “button” from the “Toolbox” and rename it as “MessageBoxButton” and Content as “Display”.
2. Double Click it to create an event for you.
3. On the c# code, insert the following code.

```
private void MessageBoxButton_Click(object sender, RoutedEventArgs e)
{
    //1. have a sting to temporary hold the data
    string myData = myInputBox.Text;
    //2. Display the data via messageBox
    MessageBox.Show(myData);
    //By default, the message box has the 'OK' button to allow the user close it
}
}
```

- 4.
5. It is also possible to add a ‘Caption’ for your message Box and handle the results of a button clicked.

```
37 private void MessageBoxButton_Click(object sender, RoutedEventArgs e)
38 {
39     //1. have a sting to temporary hold the data
40     string myData = myInputBox.Text;
41     .....
42     string message = "Display the Data on the Next Page?";
43     string caption = "Please Confirm First!";
44     .....
45     //2. Display the data via messageBox and get user input
46     MessageBoxResult result = MessageBox.Show(message, caption, System.Windows.MessageBoxButton.OKCancel);
47     .....
48     //If the user press OK, the data will be navigate to PageOne.xaml
49     if (result == MessageBoxResult.OK)
50     {
51         NavigationService.Navigate(new Uri("/PageOne.xaml?data="+myData, UriKind.Relative));
52     }
53     .....
54     }
55     .....
56 }
```

- 6.

D.Application Bar

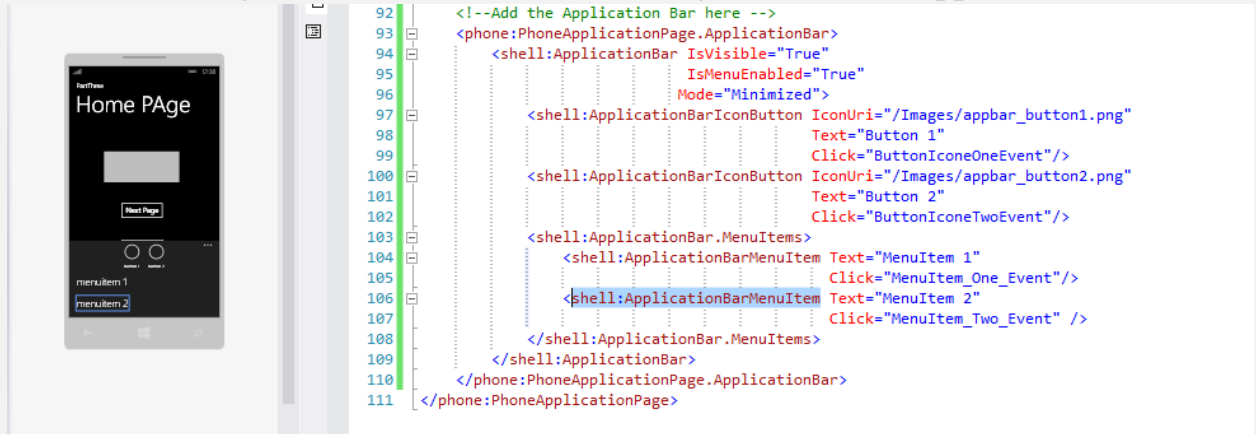
In a Windows Phone, a default Application Bar is a row of icon buttons and an ellipsis along the bottom of the phone’s screen. The ellipsis when clicked displays the labels for the menu items and icon buttons. This Bar automatically adjusts on different phone orientation; on landscape orientation the Application Bar appears on the side of the screen vertically. We can have either ‘**icon buttons**’ or ‘**Menu items**’ or even both on the application bar. By default the text label for ‘icon button; and menu label is in lowercase.

Application bar can be created using the c# code or using the .xaml which is easier to create.

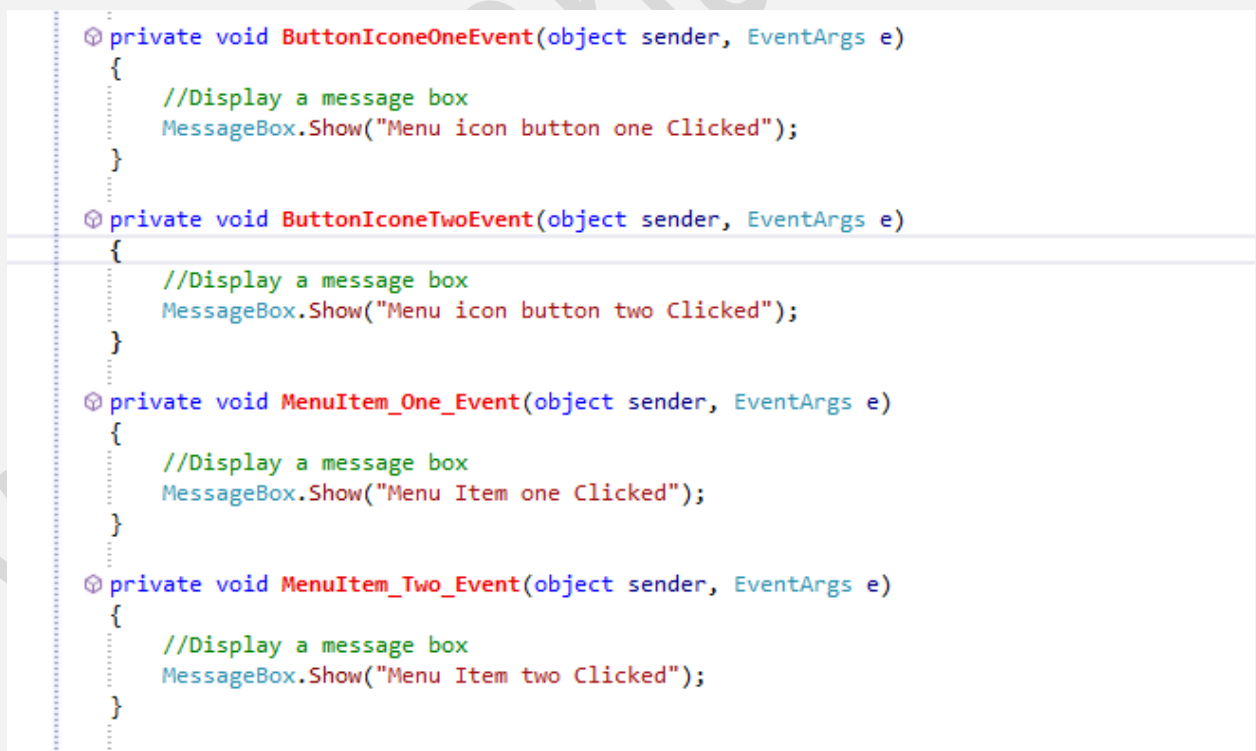
You can have the Application Bar either in two modes, "Mode="Minimized"" or "Mode="Default""

Each menu item and icon button can have its own event like shown.

On our MainPage.xaml, let's add the following code for the Application Bar.



Here is your c# code for the event created by each and every button icon and menu item.



Get the source code [here](#).

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](#)

WP8 by Bhakitah