

# Using the Windows Phone Pivot Control

The Windows Phone Pivot control is similar to a Tab control on Windows Forms, WPF or Silverlight. Each “tab” or PivotItem you create is like a separate little page where you can have whatever other XAML you need. In this short blog post I will show you how to create a set of PivotItem controls where each control (tab) will display a list box of photos. You will be able to take a set of your photos, drop them into this project and try out this sample right away. In Figure 1 you can see the sample application that you will create. I named it “My Life in Categories” and it has a list of “tabs” across the top. Each tab is a different year and in each year is a list box with a list of pictures from your life.



Figure 1: A List Box can be used in each PivotItem

## Create the Pivot Page

To start, open VS.NET 2010 and choose New Project. Click on the **Silverlight for Windows Phone** tab, then click on **Windows Phone Application** and set the name to **WPPivotSample** as shown in Figure 2. NOTE: For this particular sample,

do NOT choose the “Windows Phone Pivot Application”. We just want to create a simple pivot page and not worry about all the data stuff.

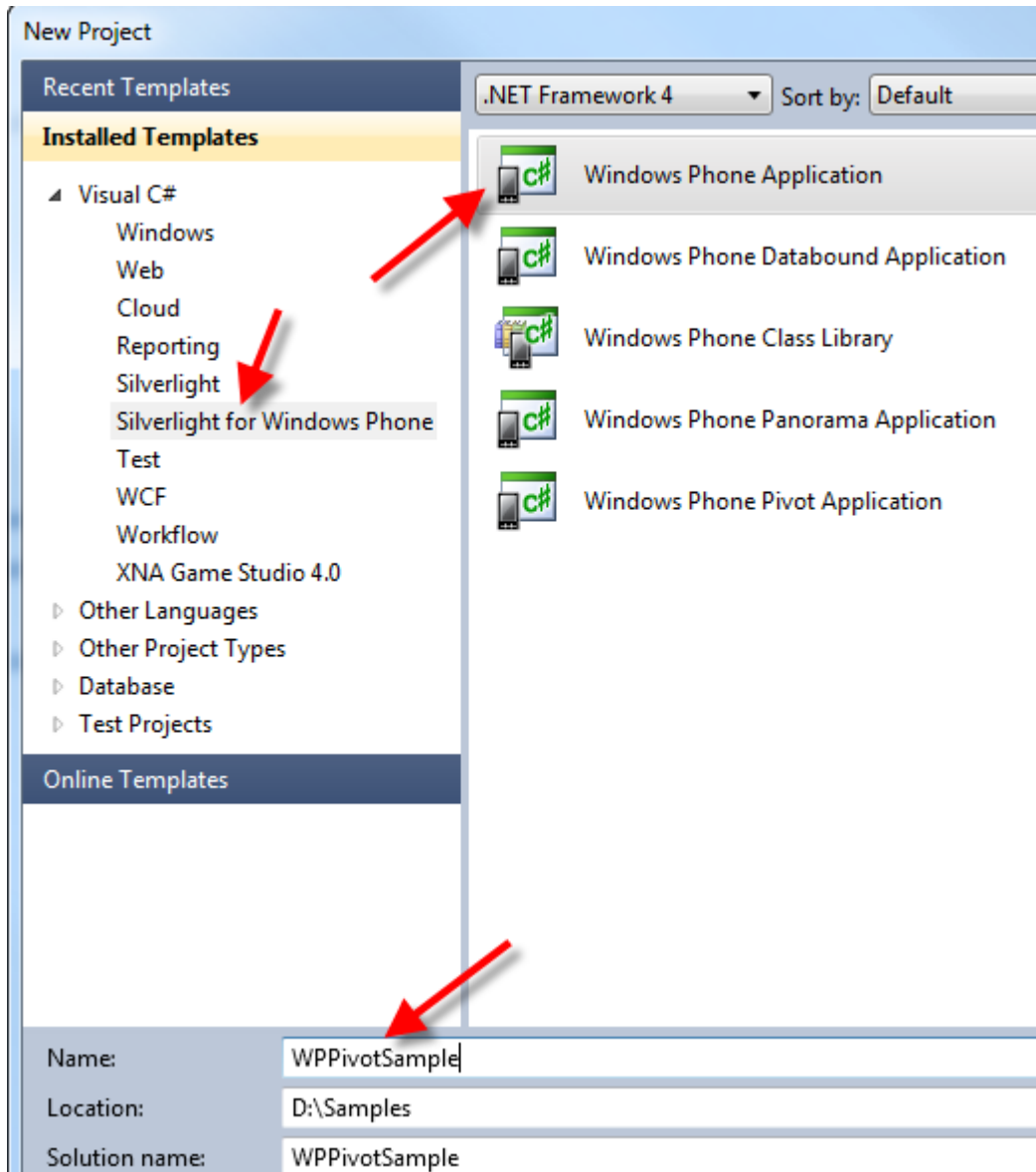


Figure 2: Create a new Windows Phone Application

Once VS.NET has your new project created, click on MainPage.xaml and delete this page. Right mouse click on the project and select Add | Add New Item... from the context menu. Select **Windows Phone Pivot Page** from the item list and set the Name to **MainPage.xaml** (Figure 3).

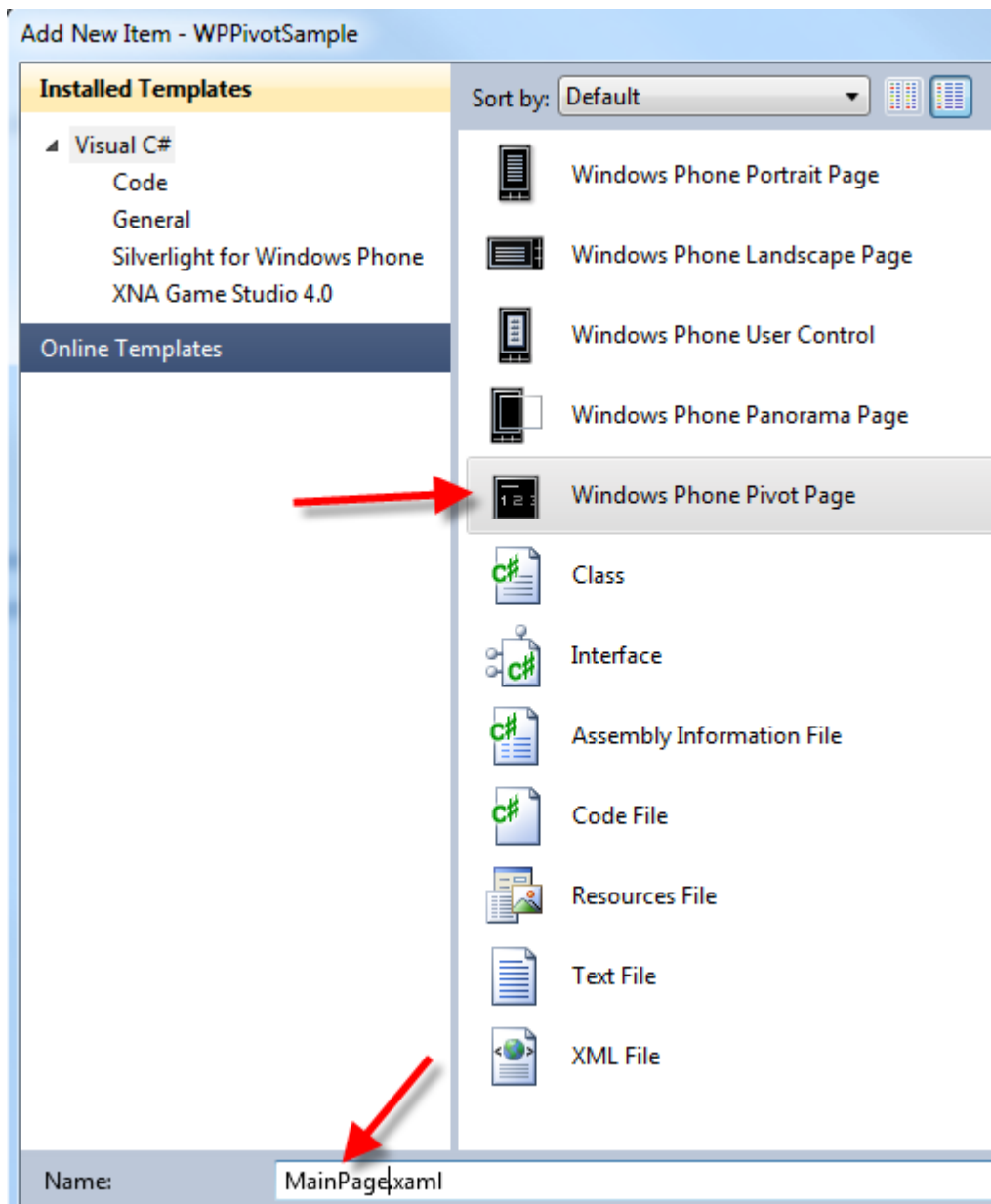


Figure 3: Add a Pivot Page

Click the OK button and your new main page will be ready for you to start creating the different Pivot Item controls in your Pivot page.

Locate the XAML for the Pivot control within this page; it will look like the following.

```
<!--Pivot Control-->
<controls:Pivot Title="MY APPLICATION">
  <!--Pivot item one-->
  <controls:PivotItem Header="item1">
    <Grid />
  </controls:PivotItem>
  <!--Pivot item two-->
  <controls:PivotItem Header="item2">
    <Grid />
  </controls:PivotItem>
</controls:Pivot>
```

A Pivot control is located in the Microsoft.Phone.Controls.dll, so an assembly reference was added to your project. An xml namespace with an alias of “controls” was added to the MainPage.xaml. This is why there is a “controls:” prefix on the Pivot control you see in the XAML.

Each Pivot control allows you to create a title that will be displayed at the top of your Pivot control. Go ahead and modify the **Title** property to “My Life in Categories”.

## Add Pictures and PivotItem Controls

You are now ready to start adding your own pictures to your project and add those pictures to various list boxes within each PivotItem control you create. The Pivot control is made up of one or many PivotItem controls. Within the content for each of these PivotItem controls you may put any XAML you want. For our purposes here you will replace the <Grid /> control with a <ListBox> control.

First, you need to create an \Images folder in your VS.NET project. Then gather any amount of pictures from your life that wish to display and copy them into this \Images folder in the project.

Now, you are ready to take a group of pictures that you want to represent one year of your life. You will create a ListBox of this set of pictures and put that ListBox into a PivotItem control. Modify the XAML in the MainPage.xaml to look like the following. Replace my image file names with your image file names and modify the Header of each PivotItem control with your appropriate headers. Keep the headers short so they are not chopped off. I find less than 12 characters works best for the Header attribute.

```
<controls:PivotItem Header="1963">
  <ListBox>
    <ListBoxItem>
      <Image Source="Images/1963-1.jpg" />
    </ListBoxItem>
    <ListBoxItem>
      <Image Source="Images/1963-2.jpg" />
    </ListBoxItem>
    <ListBoxItem>
      <Image Source="Images/1963-3.jpg" />
    </ListBoxItem>
  </ListBox>
</controls:PivotItem>

<controls:PivotItem Header="1965">
  <ListBox>
    <ListBoxItem>
      <Image Source="Images/1965-1.jpg" />
    </ListBoxItem>
    <ListBoxItem>
      <Image Source="Images/1965-2.jpg" />
    </ListBoxItem>
    <ListBoxItem>
      <Image Source="Images/1965-3.jpg" />
    </ListBoxItem>
  </ListBox>
</controls:PivotItem>
```

After putting in your images go ahead and run your application and you will be able to click on each of the headers you create to move to a new PivotItem. Then you can scroll through the ListBox on each Pivot Item to see the pictures in the list.

## Summary

A Pivot control is well suited when you have a “tab-like” metaphor you wish to convey in your application. In this sample you used a list box for each pivot item, however, each pivot item control could have a different set of XAML. Since you can create user controls in Windows Phone, each PivotItem control could actually contain a different user control. This approach would give you a great way to break up the functionality of each pivot item.

NOTE: You can download this article and many samples like the one shown in this blog entry at my website. <http://www.pdsa.com/downloads>. Select “Tips and Tricks”, then “Using the Windows Phone Pivot Control” from the drop down list.

Good Luck with your Coding,  
Paul Sheriff

**\*\* SPECIAL OFFER FOR MY BLOG READERS \*\***

We frequently offer a FREE gift for readers of my blog. Visit <http://www.pdsa.com/Event/Blog> for your FREE gift!