NOKIA Developer

HubTile in Windows Phone

This article demonstrates how to arrange HubTile as a wide Panorama item using Windows Phone Toolkit.

21 Oct 2012

Introduction

The latest version of Silverlight for Windows Phone Toolkit provides Windows Phone HubTile control. This control can make your application stylish by adding animated tiles in it. HubTile can consist of an image, title, message or a notification. In this example we will represent HubTile in two different ways, Static HubTile and Dynamic HubTile.

Static HubTile: In a Static HubTile, the properties like *Title*, *Message* and image *Source* can set from local storage which can be embedded along with the application. These properties remain and remains unchanged.

Dynamic HubTile : In a Dynamic HubTile, the properties *Title*, *Message* and image *Source* can be fetched from a remote location and updated using Data Binding technique.

Read more about HubTiles here.

Note: The Static and Dynamic HubTile are not Microsoft naming convention, these names are used by author to explain the two different ways of representing HubTile in this example.

Static HubTile

Dynamic HubTile

Implementation

Create an empty Windows Phone Project

Follow these steps.

1. Launch **Visual Studio**
2. Click on **File**
3. Create **New Project**
4. Select **Windows Phone Panorama Application** (Visual C# Template)
5. Add **Name** and **Location** of the project -> click **OK** to create the project.

Adding Windows Phone Toolkit Reference

Before you start writing code, add the reference **Microsoft.Phone.Controls.Toolkit.dll** assembly in your Windows Phone application project. And then add the following namespace declaration in XAML.

```
xmlns:toolkit="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls.Toolkit"
```

Creating wide Panorama items

To display HubTile in a wide panorama view you need add Windows Phone Toolkit `WrapPanel` control. Also, the `Orientation` property of `PanoramaItem` and `WrapPanel` should be set to `Horizontal`.

This will wind up the HubTile in a single row.

```
<controls:PanoramaItem Header="static hubtile" Orientation="Horizontal">
  <toolkit:WrapPanel x:Name="horizontalWrapPanel" Orientation="Horizontal"
```


(C) Copyright Nokia 2014. All rights reserved.

```
VerticalAlignment="Top"/>
</controls:PanoramaItem>
```

Static HubTile

We'll add 2 `StackPanel` components with horizontal orientation and appropriate margin. Thereafter, we'll add add 3 `HubTile` controls to each `StackPanel` and set its properties, as shown below.

```
<!--Panorama item one-->
  <controls:PanoramaItem Header="static hubtile" Orientation="Horizontal">
  <!-- HubTiles Start -->
  <StackPanel Orientation="Horizontal" Margin="12,0,0,0"  >
    <toolkit:WrapPanel x:Name="horizontalWrapPanel" Orientation="Horizontal"
VerticalAlignment="Top"/>
      <StackPanel Orientation="Horizontal" Margin="0,0,0,0">
         <toolkit:HubTile Title="Starter" Message="Mozza Rella"
Source="/HubTilesPanorama;component/Images/starter_mozza_rella.jpg"
HorizontalAlignment="Left" VerticalAlignment="Top" Margin="0,0,0,0"  Background="Red"
GroupTag="StaticHubTile"/>
        <toolkit:HubTile Title="Main Course" Message="Barramundi Chermula Paste"
Source="/HubTilesPanorama;component/Images/maincourse_barramundi_chermula_paste.jpg"
HorizontalAlignment="Left" VerticalAlignment="Top" Margin="12,0,0,0" Background="Red"
GroupTag="StaticHubTile"/>
        <toolkit:HubTile Title="Salad" Message="Thai Spicy Shrimp Salad"
Source="/HubTilesPanorama;component/Images/salad_thai_spicy_shrimp.jpg"
HorizontalAlignment="Left" VerticalAlignment="Top" Margin="12,0,0,0" Background="Red"
GroupTag="StaticHubTile"/>
    </StackPanel>
    <StackPanel Orientation="Horizontal" Margin="-541,185,0,0">
      <toolkit:HubTile Title="Snacks" Message="Tanzania Bacon"
Source="/HubTilesPanorama;component/Images/snacks_tanzania_bacon.jpg"
HorizontalAlignment="Left" VerticalAlignment="Top" Margin="0,0,0,0" Background="Red"
GroupTag="StaticHubTile"/>
      <toolkit:HubTile Title="Soup" Message="Potato Kale Soup"
Source="/HubTilesPanorama;component/Images/soup_potato_kale.jpg"
HorizontalAlignment="Left" VerticalAlignment="Top" Margin="12,0,0,0 " Background="Red"
GroupTag="StaticHubTile"/>
      <toolkit:HubTile Title="Beverage" Message="Water Melon Aqua Fresca"
Source="/HubTilesPanorama;component/Images/beverage_water_melon_aqua_fresca.jpg"
HorizontalAlignment="Left" VerticalAlignment="Top" Margin="12,0,0,0 " Background="Red"
GroupTag="StaticHubTile"/>
    </StackPanel>
  </StackPanel>
</controls:PanoramaItem>
```

Along with `Title`, `Message` and `Source` we set `Background` to **Red** which takes device accent color by default. These HubTile components will flip and translate animation properly. But the key properties `Title`, `Message` and `Source` would still remain static.

Dynamic HubTile

The key properties of a HubTile like `Title`, `Message`, `Source`, etc. can also be made dynamic, so that they display updated information using Data Binding concept.

To archive this, we parse an XML file and bind data to the HubTile control. This data can be from a local storage or from a remote server. In this example the XML is stored in device, however you may also fetch the XML data from a remote location.

```
//load databinding hubtile
try
{
  XElement xmlData = XElement.Load("XML/HubTileContent.xml");
  XNamespace ns = "";
  string LocalIconsPath = "Images/DataBingingImages/";
  listboxDataBinding.Items.Clear();
  foreach (var item in xmlData.Descendants("item"))
  {
    HubTileData iHubTileData = new HubTileData();
    iHubTileData.Title = (string)item.Element("title").Value;
```


(C) Copyright Nokia 2014. All rights reserved.

```
    iHubTileData.Message = (string)item.Element("message").Value;
    iHubTileData.Image = LocalIconsPath + (string)item.Element(ns + "image").Value;
    listboxDataBinding.Items.Add(iHubTileData);
  }
}
catch (System.Xml.XmlException ex)
{
   MessageBox.Show("limited connectivity or invalid data.\nplease try again");
}
```

```
<controls:PanoramaItem Header="data binding">
                <!--Double line list with image placeholder and text wrapping-->
                <ListBox Grid.Row="0" x:Name="listboxDataBinding">
                    <ListBox.ItemsPanel>
                        <ItemsPanelTemplate>
                            <toolkit:WrapPanel Orientation="Horizontal" />
                        </ItemsPanelTemplate>
                    </ListBox.ItemsPanel>
                    <ListBox.ItemTemplate>
                        <DataTemplate>
                            <toolkit:HubTile Margin="12,12,0,0"
                             Title="{Binding Title}"
                             Message="{Binding Message}"
                             Source="{Binding Image}"
                             GroupTag="BindingHubTile">
                            </toolkit:HubTile>
                        </DataTemplate>
                    </ListBox.ItemTemplate>
                </ListBox>
</controls:PanoramaItem>
```

The above XAML code will display six HubTile in a ListBox control.

Panorama Performance

Once this is done, you are ready to build and run the code to see a static and a dynamic HubTile in two Panorama items. You may observe a delay and/or a flicker while navigating the panorama items. This is due to the animation in the HubTile. To minimize this the animation of the HubTile which is not in focus can be stopped. This action is called *Freezing and Unfreezing* of HubTile. To achieve this, you can add a `GroupTag` property to all the HubTile controls in a XAML file. `FreezeGroup()` and `UnfreezeGroup()` methods can be called to stop and start the HubTile animation, respectively.

```
private void Panorama_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
  if (Panorama_HubTile.SelectedIndex == 0)
  {
  HubTileService.UnfreezeGroup("StaticHubTile");
  HubTileService.FreezeGroup("BindingHubTile");
  }
  else if (Panorama_HubTile.SelectedIndex ==1)
  {
  HubTileService.UnfreezeGroup("BindingHubTile");
  HubTileService.FreezeGroup("StaticHubTile");
  }
}
```

Usually while launching an application with multiple numbers of panaroma items, a delay is observed. The is because the background image takes time while loading whereas the rest of the interface appears immediately. To overcome this delay, the following code can be added (while creating the background image).

```
<controls:Panorama.Background>
                <ImageBrush>
                    <ImageBrush.ImageSource>
                        <BitmapImage CreateOptions="DelayCreation"
UriSource="lightbg.jpg"></BitmapImage>
                    </ImageBrush.ImageSource>
```


(C) Copyright Nokia 2014. All rights reserved.

```
                </ImageBrush>
</controls:Panorama.Background>
```

`CreateOptions=DelayCreation` property loads the background image when the app launches. This will give a smooth and fluid panorama background animation. To learn more about this, please see Silverlight for Windows Phone Performance.

Summary

This example shows the use of HubTile control in wide panorama item, suggests workarounds better performance. The application is set to light theme. To set a particular theme in your application see PhoneThemeManager

Source Code

The complete source code of the example is available here: File:HubTilesPanorama.zip

(C) Copyright Nokia 2014. All rights reserved.