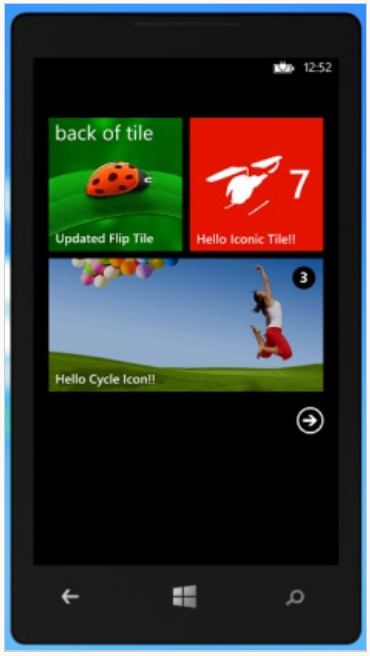
Live Tile Templates in Windows Phone 8

This article explains how to use Windows Phone 8 Live Tiles templates API. Windows Phone 8 has introduced new tile features such as multiple sizes, customizable, specific page navigation using secondary tile, static and dynamic updates etc.

Introduction

In simple terms, Tile is a shortcut of an application, which is shown on the start screen, also known as a primary tile or default tile. An application can also have one or more additional tiles, known as secondary tiles.



Windows Phone 8 Live Tile

What is a Live Tile?

Live tile is a part of a notification.It is a smart, simple, robust and user friendly alert system which allows you to show quick updates on your phone start screen.

Live Tiles Templates

Windows Phone 8 has introduced three types of Live Tiles Templates with different sizes.

1. FlipTileData
2. CycleTileData
3. IconicTileData

These tile templates use three standard sizes, which are small, medium, and wide.

Name of Tiles	Small (Pixels)	Medium (Pixels)	Wide (Pixels)
Flip	159 x 159	336 x 336	691 x 336
Cycle	159 x 159	336 x 336	691 x 336
Iconic	110 x 110	202 x 202	N/A

In this example I have built each tile programmatically using C# code

Let's create a new Windows Phone 8 project and add three buttons to the main page (MainPage.xaml) in order to develop a Flip, Cycle and iconic tile templates. See the following XAML code snippet.

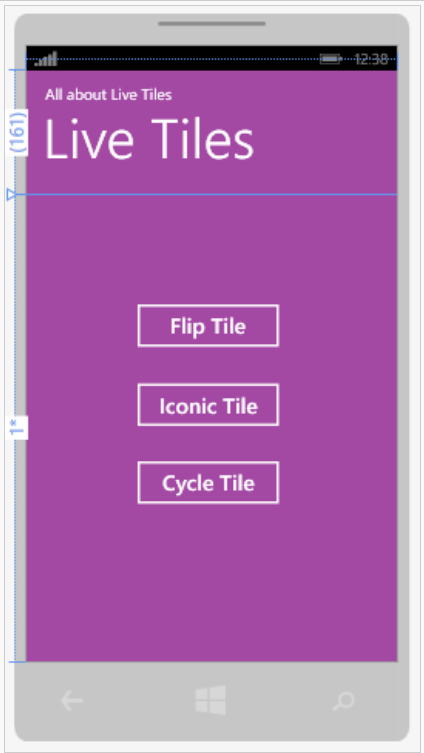
XAML code snippet

```
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
             <Button Name="btnFlipTile" Content="Flip Tile" HorizontalAlignment="Left"
Margin="120,131,0,0" VerticalAlignment="Top" Width="207" Click="btnFlipTile_Click"
```


(C) Copyright Nokia 2014. All rights reserved.

```
FontSize="28"/>
            <Button Name="btnIconicTile" Content="Iconic Tile"
HorizontalAlignment="Left" Margin="120,234,0,0" VerticalAlignment="Top" Width="207"
Click="btnIconicTile_Click"  FontSize="28"/>
            <Button Name="btnCycleTile" Content="Cycle Tile" HorizontalAlignment="Left"
Margin="120,335,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.533,2.139"
Width="207"  FontSize="28" Click="btnCycleTile_Click"/>
        </Grid>
```

UI Design View

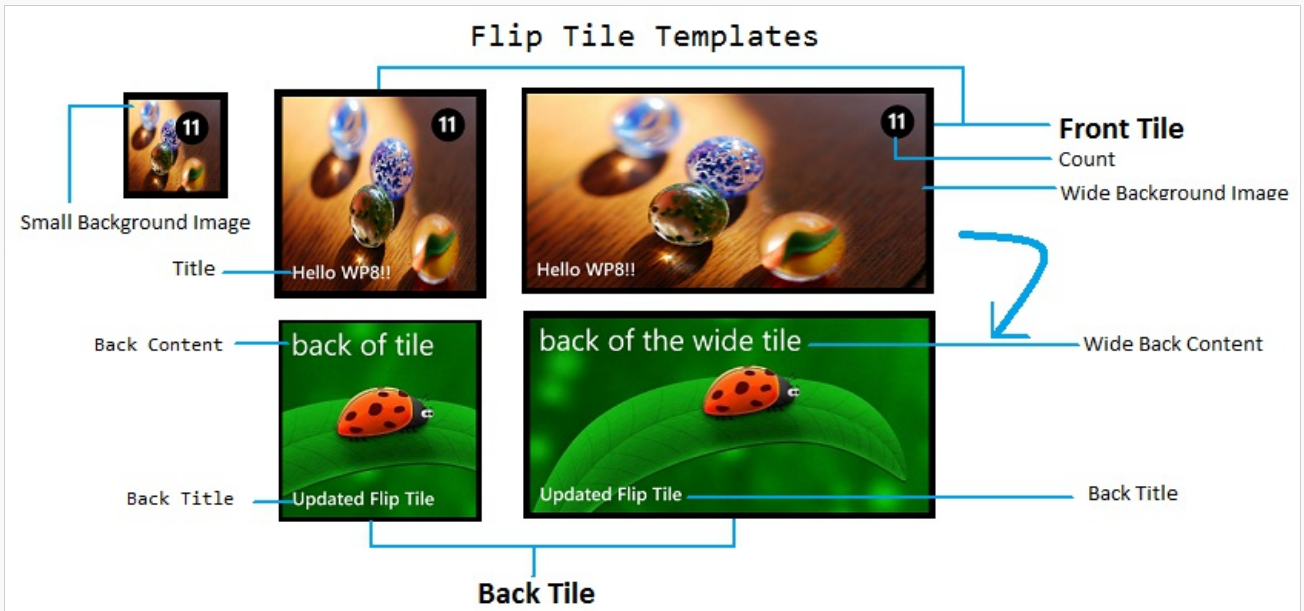


UI (User Interface) Design View

Note: You can create or update Live Tiles using either XML code or C# code. In this example I have considered C# code.

Flip Tile Template

Flip tile is also known as a primary or default tile.The phone user has to pin it form application list or from within the application. Flip tile flips from front to back and back to front. The figure shows the structure and properties of flip tiles.



Flip Tile Template

(C) Copyright Nokia 2014. All rights reserved.

Flip tiles are also supported in Windows Phone 7. To create a Flip Tile, developer needs to set few basic properties such as title, count, a back ground image and a text located on the back of the tile of `FlipTileData` class. Here, `CreateFlipTileData()` is invoked when the primary tile is not created on the start screen .Rest of the code deals with updating the tile with data.

```
        private void btnFlipTile_Click(object sender, RoutedEventArgs e)
         {
             // find the tile object for the application tile that using "flip" contains
string in it.
             ShellTile oTile = ShellTile.ActiveTiles.FirstOrDefault(x =>
x.NavigationUri.ToString().Contains("flip".ToString()));


             if (oTile != null && oTile.NavigationUri.ToString().Contains("flip"))
             {
                 FlipTileData oFliptile = new FlipTileData();
                 oFliptile.Title = "Hello WP8!!";
                 oFliptile.Count = 11;
                 oFliptile.BackTitle = "Updated Flip Tile";

                 oFliptile.BackContent = "back of tile";
                 oFliptile.WideBackContent = "back of the wide tile";

                 oFliptile.SmallBackgroundImage = new
Uri("Assets/Tiles/Flip/159x159.png", UriKind.Relative);
                 oFliptile.BackgroundImage = new Uri("Assets/Tiles/Flip/336x336.png",
UriKind.Relative);
                 oFliptile.WideBackgroundImage = new Uri("Assets/Tiles/Flip/691x336.png",
UriKind.Relative);

                 oFliptile.BackBackgroundImage = new Uri("/Assets/Tiles/Flip/A336.png",
UriKind.Relative);
                 oFliptile.WideBackBackgroundImage = new
Uri("/Assets/Tiles/Flip/A691.png", UriKind.Relative);
                 oTile.Update(oFliptile);
                 MessageBox.Show("Flip Tile Data successfully update.");
             }
             else
             {
                 // once it is created flip tile
                 Uri tileUri = new Uri("/MainPage.xaml?tile=flip", UriKind.Relative);
                 ShellTileData tileData = this.CreateFlipTileData();
                 ShellTile.Create(tileUri, tileData, true);
             }
         }

         private ShellTileData CreateFlipTileData()
         {
             return new FlipTileData()
             {
                 Title = "Hi Flip Tile",
                 BackTitle = "This is WP8 flip tile",
                 BackContent = "Live Tile Demo",
                 WideBackContent = "Hello Nokia Lumia 920",
                 Count = 8,
                 SmallBackgroundImage = new Uri("/Assets/Tiles/Flip/A159.png",
UriKind.Relative),
```


(C) Copyright Nokia 2014. All rights reserved.

```
                BackgroundImage = new Uri("/Assets/Tiles/Flip/A336.png",
UriKind.Relative),
                WideBackgroundImage = new Uri("/Assets/Tiles/Flip/A691.png",
UriKind.Relative),
            };
        }
```

First of all, we have to identify whether the application tile is available or not on the start screen, using `ShellTile` API class. Using `ShellTile` API we can get the current active tile by query. After that, we will instantiate a new object of `FlipTileData` and set the following properties. After setting the properties, we need to call the `Update()` method of `ShellTile` object and it will update the primary tile.

```
oFliptile.Title = "Hello WP8!!";
oFliptile.Count = 11;
oFliptile.BackTitle = "Updated Flip Tile";
oFliptile.BackContent = "back of tile";
oFliptile.WideBackContent = "back of the wide tile";
oFliptile.SmallBackgroundImage = new Uri("Assets/Tiles/Flip/159x159.png",
UriKind.Relative);
oFliptile.BackgroundImage = new Uri("Assets/Tiles/Flip/336x336.png", UriKind.Relative);
oFliptile.WideBackgroundImage = new Uri("Assets/Tiles/Flip/691x336.png",
UriKind.Relative);
oFliptile.BackBackgroundImage = new Uri("/Assets/Tiles/Flip/A336.png",
UriKind.Relative);
oFliptile.WideBackBackgroundImage = new Uri("/Assets/Tiles/Flip/A691.png",
UriKind.Relative);
```

Cycle Tile Template

Cycle tile is similar to a windows phone 7.5's pictures live tile.Cycle tile was not possible via code in Windows Phone 7.5, but it is now possible to create that in Windows Phone 8 using the `CycleTileData` class.lit could be a secondary tile of your application. The cycle tile template API allows you to cycles one to nine images in random time interval. Cycle tile does not have back of a tile. Cycle tile have three standard size ; small, medium and wide. Small size tile does not cycle any images, however you can set a static background image, otherwise it will take phone theme color by default.Below figure shows the structure and properties of flip tiles.

Cycle Tile Template

Here is `CaycleTileData` code snippet. It is invoked when user clicks on Cycle Tile button.

```
private void btnCycleTile_Click(object sender, RoutedEventArgs e)
        {
            CycleTileData oCycleicon = new CycleTileData();
```


(C) Copyright Nokia 2014. All rights reserved.

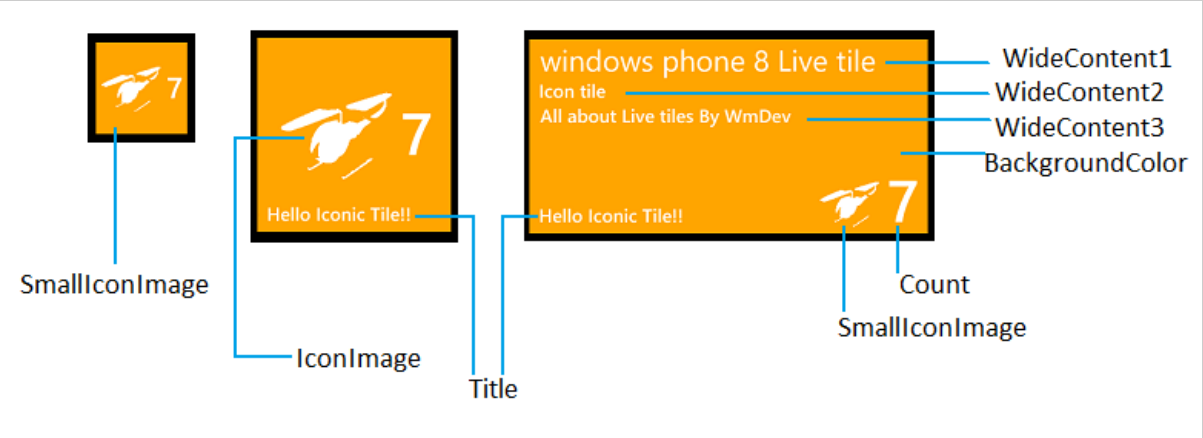
```
            oCycleicon.Title = "Hello Cycle Icon!!";
            oCycleicon.Count = 9;

            oCycleicon.SmallBackgroundImage = new Uri("Assets/Tiles/Cycle/159x159.png",
UriKind.Relative);

            // Images could be max Nine images.
            oCycleicon.CycleImages = new Uri[]
               {
                  new Uri("Assets/Tiles/Cycle/001.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/002.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/003.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/004.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/005.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/006.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/007.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/008.jpg", UriKind.Relative),
                  new Uri("Assets/Tiles/Cycle/009.jpg", UriKind.Relative),
               };

            // find the tile object for the application tile that using "cycle" contains
string in it.
            ShellTile TileToFind = ShellTile.ActiveTiles.FirstOrDefault(x =>
x.NavigationUri.ToString().Contains("cycle".ToString()));

            if (TileToFind != null &&
TileToFind.NavigationUri.ToString().Contains("cycle"))
            {
                TileToFind.Delete();
                ShellTile.Create(new Uri("/MainPage.xaml?id=cycle", UriKind.Relative),
oCycleicon, true);
            }
            else
            {
                ShellTile.Create(new Uri("/MainPage.xaml?id=cycle", UriKind.Relative),
oCycleicon, true);
            }
        }
```

`ShellTile` API class allows developer to create a Cycle Tile programmatically and navigate to tiles screen.

Iconic Tile Template

The iconic tile template is a fresh template, that is pioneered in Windows Phone 8. Iconic tile itself is comprehensive. The Iconic Tile template requires two icons as transparent images such as `SmallIconImage` (110 x 110 in size) and `IconImage` (202 x 202 in size).It also supports scalability across resolutions. Iconic tile API allows you to set background color.If there is no color set, it will take automatically use the phone theme color.

(C) Copyright Nokia 2014. All rights reserved.



Iconic Tile Template (Image courtesy of Microsoft)

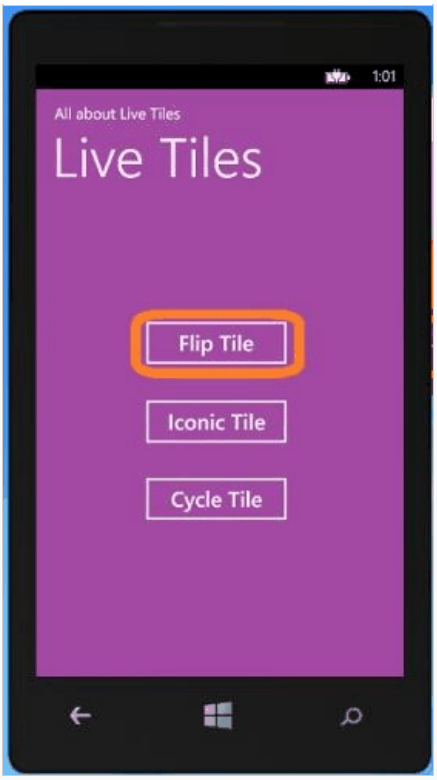
Here is `IconicTileData` code snippet.It is invoked when user clicks on Iconic Tile button.

```
        private void btnIconicTile_Click(object sender, RoutedEventArgs e)
         {
             IconicTileData oIcontile = new IconicTileData();
             oIcontile.Title = "Hello Iconic Tile!!";
             oIcontile.Count = 7;

             oIcontile.IconImage = new Uri("Assets/Tiles/Iconic/202x202.png",
UriKind.Relative);
             oIcontile.SmallIconImage = new Uri("Assets/Tiles/Iconic/110x110.png",
UriKind.Relative);

             oIcontile.WideContent1 = "windows phone 8 Live tile";
             oIcontile.WideContent2 = "Icon tile";
             oIcontile.WideContent3 = "All about Live tiles By WmDev";

             oIcontile.BackgroundColor = System.Windows.Media.Colors.Orange;

             // find the tile object for the application tile that using "Iconic"
contains string in it.
             ShellTile TileToFind = ShellTile.ActiveTiles.FirstOrDefault(x =>
x.NavigationUri.ToString().Contains("Iconic".ToString()));

             if (TileToFind != null &&
TileToFind.NavigationUri.ToString().Contains("Iconic"))
             {
                 TileToFind.Delete();
                 ShellTile.Create(new Uri("/MainPage.xaml?id=Iconic", UriKind.Relative),
oIcontile, true);
             }
             else
             {
                 ShellTile.Create(new Uri("/MainPage.xaml?id=Iconic", UriKind.Relative),
oIcontile, true);
             }
         }
```

Output

Here is the output of this sample code

(C) Copyright Nokia 2014. All rights reserved.



output

Download

Source code is here File:WP8 LiveTiles.zip

~ Happy Coding ~

References (MSDN)

- Tiles for Windows Phone

(C) Copyright Nokia 2014. All rights reserved.