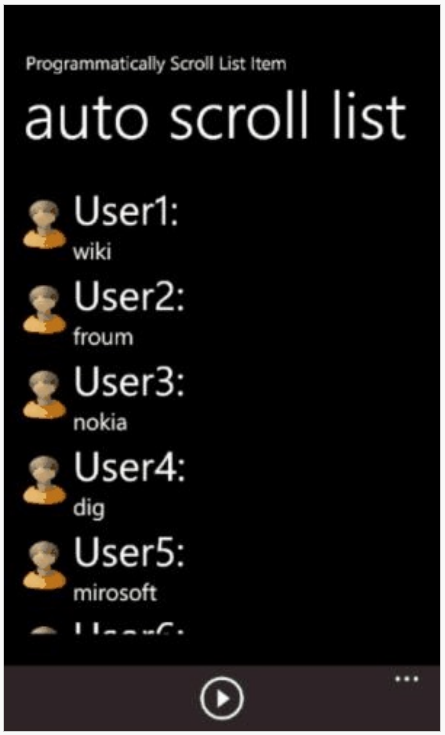
Auto-scrolling ListBox for Windows Phone

This code example explains how to create a ListBox which automatically scrolls through and highlights its items.

Introduction

Auto scrolling lists are not only useful for displaying informations that is regularly updated, but also where some meaningful information may be displayed. It can be a pattern, trend or some historical information (for example, stock market updates, data feeds, flight information display systems, social network status/data update etc).

The auto-scrolling `ListBox` illustrated in this article is displayed in the animated gif image below. The box navigates to each item in turn (ensuring it is visible) using ListBox:ScrollIntoView() and highlights the item by setting it as *Selected*. At the end of the list, the selection returns to the first item and repeats the iteration. A DispatcherTimer is used to control the rate of movement through the items.



Auto scrolling list

Architectural overview

The main components of this solution are listed below:

1. Custom List Control (XAML) - definition of the List and appearance of its list items in XAML
2. Sample List Data - C# class defining the *format* of data to be displayed in the `ListBox` along with instances of the class that are bound to the `ListBox`
3. Dispatcher Timer API - C# code for iterating through the list items.

These are discussed in the following sections

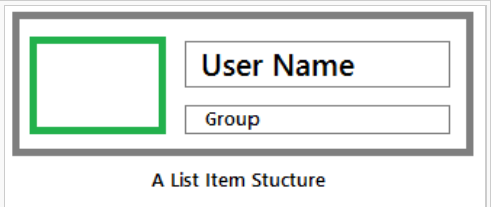
Custom List Box definition (in XAML)

The XAML definition for the `ListBox` is given below:

```
<!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
            <StackPanel Orientation="Horizontal">
                <ListBox x:Name="lstSample">
                    <ListBox.ItemTemplate>
                        <DataTemplate>
                            <StackPanel Orientation="Horizontal">
                                <Image Source="{Binding ImagePath}" Stretch="None"/>
                                <StackPanel Orientation="Vertical">
                                    <TextBlock Text="{Binding Name}" FontSize="45"/>
                                    <TextBlock Text="{Binding Group}" FontSize="25"/>
                                </StackPanel>
                            </StackPanel>
                        </DataTemplate>
                    </ListBox.ItemTemplate>
                </ListBox>
            </StackPanel>
        </Grid>
```


(C) Copyright Nokia 2013. All rights reserved.

The `ListBox` has the name `lstSample`, and is defined in terms of a DataTemplate. The template describes the appearance of each item and includes named bindings for each element of those items. In this example, a list component item will contain an image, user name and group as shown below:



List item template

Sample List Data

The sample list data is defined in the C# code behind, as shown below. Note that as required in the XAML binding definition above the properties are named `Name`, `Group`, and `ImagePath`,

```
 public class SampleData
    {
        public string ImagePath { get; set; }
        public string Name { get; set; }
        public string Group { get; set; }
    }
```

We then create instances of this data class for the list items and set them as the source for the `ListBox`.

```
//SimpleItems is a list containing items of class SampleData
SimpleItems = new List<SampleData>()
{
 new SampleData(){Name = "User1:", ImagePath="/Images/user.png", Group ="wiki"},
 new SampleData(){Name = "User2:", ImagePath="/Images/user.png", Group="froum"},
 new SampleData(){Name = "User3:", ImagePath="/Images/user.png", Group ="nokia"}
};

//Bind the list data to the ListBox (note that lstSample is the name of the list defined in the XAML)
this.lstSample.ItemsSource = SimpleItems;
```

Dispatcher Timer API

Finally we will call the DispatcherTimer API to iterate through the items.

```
// DispatcherTimer object declaration in constructor
timer = new DispatcherTimer();
timer.Interval = TimeSpan.FromMilliseconds(1500); //delayed for 1.5 seconds
timer.Tick += new EventHandler(timer_Tick);
```

When the timer is triggerd, we call the `timer_Tick` event handler, which scrolls to the current index and marks it as selected, and then updates the index. After the last item is highlighted the index is reset to the first item.

```
void timer_Tick(object sender, EventArgs e)
        {
            lstSample.ScrollIntoView(lstSample.Items[index]); //scroll to the current item
            lstSample.SelectedIndex = index; //highlight the selected item in the list box scroller
            SampleData item = (SampleData)lstSample.Items[index]; // getting the current item

            if (index < lstSample.Items.Count - 1)
            {
                index++;
            }
            else
            {
                lstSample.ScrollIntoView(lstSample.Items[0]); //scroll to the first item
                index = 0; //reset the index when it reaches to the last item
            }
        }
```

Run it using F5 key.

References

1. ScrollIntoView()
2. List<T> Class
3. DispatcherTimer Class
4. Application Bar for Windows Phone

~ Happy Coding ~

Sample Code

You can download sample code from Media:WmDevAutoScrollListBox.zip.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.