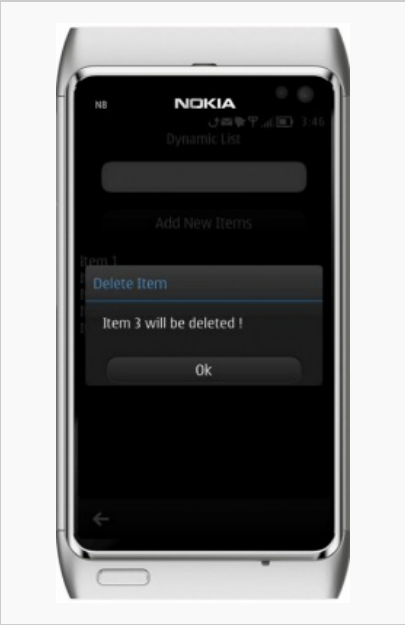
Dynamic List Item in Qt and WP7

This article demonstrates how to add/remove Dynamic List Items in Qt and WP7.

Introduction

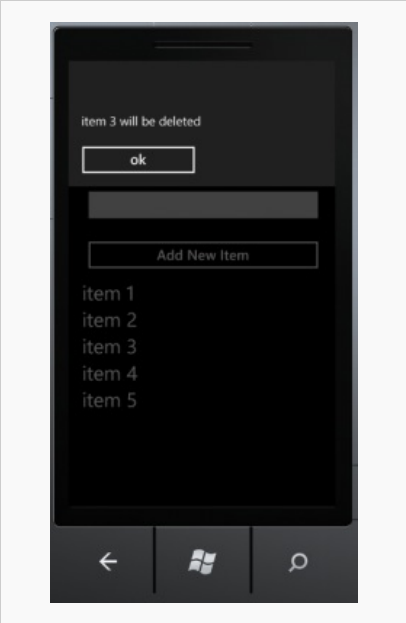
In this article we will create a Dynamic List in both Qt and WP7, where user can add and delete list items from the list box dynamically. We will add a text input box to enter the item name, a button to add the item in the list box and to delete any item user can select the item from the list.

thumb|201|Dynamic List in QML



thumb|201|Deleting Item in QML

thumb|201|Dynamic List in WP7



thumb|201|Deleting Item in WP7

Implementation

Let's create an empty project for both Qt and WP7. First we will work on Qt project and then will move on to WP7 project.

Qt Project (MainPage.qml)

Let's create a text box input using TextArea Element. This is been created to get user input and to add the text in the ListView.

(C) Copyright Nokia 2013. All rights reserved.

```
TextArea{
        id:textAreaInput
        width: parent.width-60
        anchors.horizontalCenter: parent.horizontalCenter
        anchors.top:textHeader.bottom
        anchors.topMargin:20
    }
```

We add a button using ToolButton Element. When user clicks on the ToolButton it first checks whether the TextArea is empty or not and if it's not empty then it append the text field from the TextArea into the ListModel.

```
ToolButton{
        id:buttonAdd
        text: "Add New Items"
        anchors.top: textAreaInput.bottom
        anchors.topMargin: 20
        anchors.horizontalCenter: parent.horizontalCenter
        width: parent.width-60
        onClicked: {
            if(textAreaInput !== "")
            {
            listModelItem.append({ "name":textAreaInput.text})
            textAreaInput.text= "";
            }
        }
    }
```

The method `append()` of the ListModel Element appends the text filed at the end of the ListView.

```
ListView {
         id:listviewItems
         width: parent.width;
         height: parent.height - 80
         anchors.top: buttonAdd.bottom
         anchors.topMargin: 20
         anchors.horizontalCenter: parent.horizontalCenter
         clip: true;
         Component {
             id: delegateItems
             Item {
                 id: wrapper
                 width: parent.width -20
                 height: textItem.height
              Column{
                 Text {
                     id: textItem
                     text: name
                     color:"white"
                     font.pixelSize: 21
                 }
                }
                 MouseArea{
                     id:mouseAreaItem
                     anchors.fill: parent
                     onClicked:
                     {
                      dialogItem.open();
                      listIndex= index;
                      dialogName = name;
                     }
                 }
             }
         }
         model: listModelItem
         delegate: delegateItems
         focus: true
     }
```

When user clicks on the items it brings a Dialog to notify user that the item will be deleted.

```
    CommonDialog{
        id: dialogItem
        titleText: "Delete Item"
        content:
            Item{
            anchors.left: parent.left
            anchors.leftMargin: 25
            anchors.top:parent.top
            anchors.topMargin: 20
            Text{
            text:dialogName+" will be deleted !"
            color: "white"
            font.pixelSize: 21
            }
        }
        buttons: ToolButton{
            width:parent.width-60
            anchors.horizontalCenter: parent.horizontalCenter
            text: "Ok"
            onClicked:
            {
                listModelItem.remove(listIndex);
                dialogItem.close();
            }
        }
    }
```

Which finally calls the `remove()` method of ListModel to delete the specific item from the ListView.

WP7 Project (MainPage.xaml)

Similarly like Qt we add a text box input using TextBox, a button using Button and a list box using ListBox control in the XAML page.

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
```


(C) Copyright Nokia 2013. All rights reserved.

```
            <TextBox Height="72" HorizontalAlignment="Left" Margin="12,30,0,0" Name="textBox1" Text="" VerticalAlignment="Top" Width="
            <Button Content="Add New Item" Height="72" HorizontalAlignment="Left" Margin="12,120,0,0" Name="button1" VerticalAlignme
            <ListBox Height="403" HorizontalAlignment="Left" Margin="12,198,0,0" Name="listBox1" VerticalAlignment="Top" Width="437"
</Grid>
```

Like Qt user can add some text in the TextBox area and when the user press the Button first it checks whether the text file is empty or not and then adds the text field into the ListBox as a list item.

```
private void button1_Click(object sender, RoutedEventArgs e)
        {
            if (textBox1.Text != "")
            {
                listBox1.Items.Add(textBox1.Text);
                textBox1.Text = "";
            }
        }
```

The `listBox1.Items.Add()` method of the ListBox adds item in the list. When user select any item from the ListBox it display a notification to the user that the item will be deleted and then the item gets deleted from the ListBox.

```
private void listBox1_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {
            if (e.AddedItems.Count > 0)
            {
               MessageBox.Show(listBox1.SelectedItem.ToString()+" will be deleted");
               listBox1.Items.RemoveAt(listBox1.Items.IndexOf(listBox1.SelectedItem));
               ((ListBox)sender).SelectedIndex = -1;
            }
        }
```

When user select any item from the ListBox the `listBox1_SelectionChanged()` event handler gets called and finds *index* of the selected item which then calls the `listBox1.Items.RemoveAt()` method to delete the specific item from the ListBox.

Source Code

- The full source code of Qt example is available here: File:DynamicListModelQML.zip
- The full source code of WP7 example is available here: File:DynamicListModelWP7.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.