How to use Auto Complete Box in Windows Phone 7

This article shows how to use AutoCompleteBox, a Windows Phone 7 control that provides a text box for user input and a drop-down that contains possible matches based on the input in the text box.

Pre-requisites

The AutoCompleteBox is defined in the Silverlight Toolkit, whose most recent version you can find here. If the toolkit is not installed already, please follow the installation instructions below.

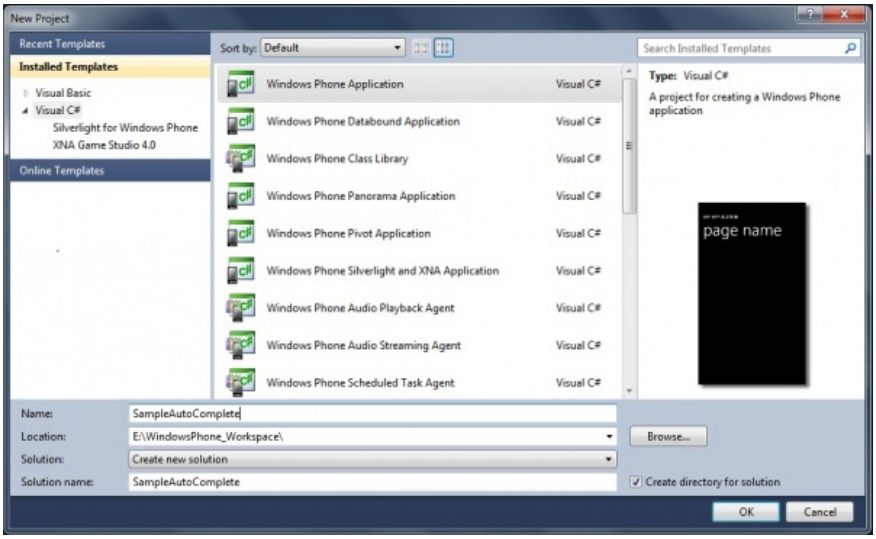
Installing the Silverlight Toolkit

- Installation of the most recent Siverlight Toolkit needs an installation of NuGet.
- After NuGet is installed, the Toolkit has to be installed from the NuGet Package Manager Console in the Tools menu / Library Packge Manger submenu in Visual Studio.
- Type "Install-Package SilverlightToolkitWP" there.

The toolkit should be installed into your current solution. On my machine with a preinstalled elder version (Oct 2011) of the toolkit I had problems. Removing the reference to the old assembly and finding the new one in the packages folder of the solution solved the problems on one machine but not on another...

Starting a New Project

- Create a new project in Visual Studio 2010 from File>>New Project>>Windows Phone Application - Visual C#. Give a desired name of the project.



(C) Copyright Nokia 2013. All rights reserved.

Using IEnumerable

- We now need to make a class derived from `IEnumerable` interface. For this, right click the project in Project Explorer window: >>Add>>New Item>>Class (Here, we are using 'SampleWords' as our class name).
- If we derive any class from IEnumerable interface, then it's compulsory to define the function `GetEnumerator()`. So, our class should look like below:

```
namespace SampleAutoCompleteBox
{
    public class SampleWords : IEnumerable
    {
        public IEnumerable AutoCompletions = new List<string>()
        {
         "Lorem", "ipsum", "dolor", "sit", "amet", "consectetur", "adipiscing", "elit", "Nullam", "felis", "dui", "gravida", "
         "condimentum", "eget", "mattis", "non", "est", "Duis", "porta", "ornare", "tellus", "at", "convallis", "nibh", "aliqu
         "faucibus", "Vivamus", "molestie", "fringilla", "ullamcorper", "Aenean", "non", "diam", "eu", "sapien", "pretium", "i
         "Quisque", "at", "ante", "libero", "eu", "tincidunt", "urna", "Cras", "libero", "ligula", "hendrerit", "at", "posuere
         "tempor", "at", "nulla", "Aliquam", "feugiat", "sagittis", "dolor", "convallis", "porttitor", "neque", "commodo", "ut
         "egestas", "tincidunt", "lectus", "et", "pharetra", "enim", "semper", "et", "Fusce", "placerat", "orci", "vel", "iacu
         "dictum", "nulla", "sem", "convallis", "nunc", "in", "viverra", "leo", "mauris", "eu", "odio", "Nullam", "et", "ultri
         "sapien", "Proin", "quis", "mi", "a", "sapien", "semper", "lobortis", "ut", "eget", "est", "Suspendisse", "scelerisqu
         "mattis", "In", "eleifend", "tellus", "vel", "nulla", "aliquam", "ornare", "Praesent", "tincidunt", "dui", "ut", "lib
         "iaculis", "consequat", "Nunc", "interdum", "eleifend", "rhoncus", "Curabitur", "sollicitudin", "nulla", "sagittis",
         "vehicula", "cursus", "Fusce", "laoreet", "arcu", "vitae", "fringilla", "scelerisque", "nisi", "purus", "laoreet", "i
         "suscipit", "erat", "tellus", "eu", "sapien", "Proin", "pharetra", "tortor", "nisl", "Etiam", "et", "risus", "eget",
         "vulputate", "dignissim", "ac","sed", "erat", "Nulla", "vel", "condimentum", "nunc", "Suspendisse", "aliquam", "euism
         "Ut", "arcu", "enim", "consectetur", "at", "rhoncus", "at", "porta", "ut", "lacus", "Donec", "nisi", "quam", "faucibu
         "tincidunt", "eu", "porttitor", "id", "ipsum", "Proin", "nec", "neque", "nulla", "Suspendisse", "sapien", "metus", "a
         "dapibus", "consequat", "rutrum", "id", "leo", "Donec", "ac", "fermentum", "tortor", "Pellentesque", "nisl", "orci",
         "at", "iaculis", "vitae", "consequat", "scelerisque", "ante", "Suspendisse", "potenti", "Maecenas", "auctor", "justo"
         "sagittis", "facilisis", "Phasellus", "ultrices", "lectus", "a", "nisl", "pretium", "accumsan"
         };

        public IEnumerator GetEnumerator()
        {
            return AutoCompletions.GetEnumerator();
        }
    }
}
```

- Remember, not to forget adding the following 2 using directives

```
using System.Collections; // for 'IEnumerable'
using System.Collections.Generic; // for 'List'
```

- Thus, we have declared a list of strings as our sample strings. Now to work on the UI.

Making UI and attaching Auto Complete Box with SampleWords class

- We have to include the namespace inside which we have created the Strings class (SampleWords class according to this example) in the .xaml file in which we need to put `AutoComplete` box.
- We will add below line of code in the MainPage.xaml (e.g)

```
xmlns:local="clr-namespace:SampleAutoCompleteBox"
```

The `xmlns:local` always point to the top level namespace. So, we have defined **SampleWords.cs** inside the `SampleAutoCompleteBox` namespace (same as your project name).

- Create/declare a static resource in the MainPage.xaml file like shown below

```
<phone:PhoneApplicationPage.Resources>
    <local:SampleWords
    x:Key="AutoCompletions" />
</phone:PhoneApplicationPage.Resources>
```

Here, 'local:' will have the class file name in which we have declared the list of stings. We have also defined a key as 'AutoCompletions' so that it can be directly used by it in future.

- Now we will drag and drop the `AutoCompleteBox` tool from the Toolkit on to the MainPage design view. By this, the corresponding code will be inserted into the code view.
- We will assign the value `StaticResource AutoCompletions` to the property `ItemsSource` as shown below:

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
      <toolkit:AutoCompleteBox HorizontalAlignment="Left"  Width="450" Grid.Row="0"
                               Name="autoCompleteBox1" VerticalAlignment="Top"
                               ItemsSource="{StaticResource AutoCompletions}"/>
</Grid>
```

- Now build the project and run it to see AutoCompleteBox tool running.

Source code

- The example illustrated in the article can be downloaded from here .

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.