A201/A505 Laboratory Session #6

LAB GOALS

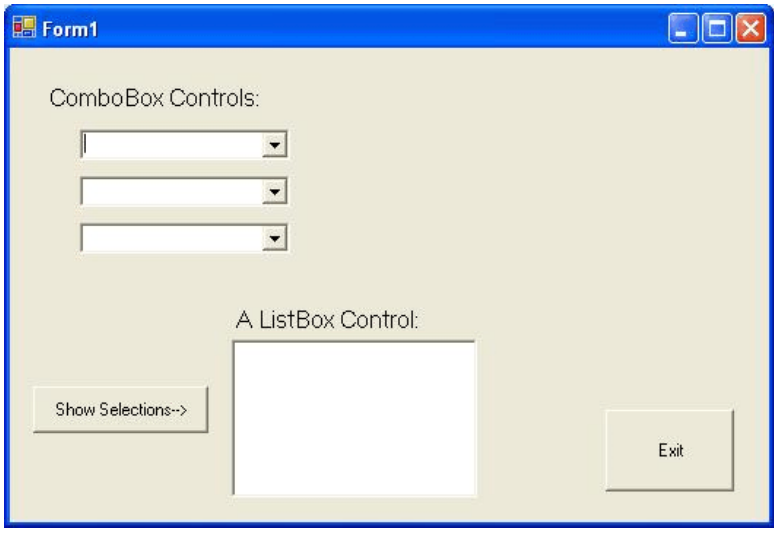
To learn how to use advanced VB controls such as Combo Boxes and List Boxes. We will examine the properties, methods and event handlers for these two objects.

In addition, we will review visual basic=s logical constructs such as the IF-THEN-ELSE.

To get started, proceed to open up Microsoft Visual Basic .NET.

Step 1: Create a new project named “ComboBox”. Once the main form is displayed, create the following interface. Make sure to save your new project in your A201 or A505 folder.

2 Buttons, 1 label, 1 listbox, and 3 combobox.




```
ComboBox:    From the top:
             ComboBoxMake,
             ComboBoxModel,
             ComboBoxColor

Listbox:     ListBoxOutput

Button:      BtnShowSelection,
             BtnExit
```

Step 2: Double click the Exit button and write the code that allows your program to properly exit.

To invoke strong type checking by the compiler, don’t forget to add the following two lines at the beginning of the program:

```
Option Explicit On
Option Strict On
```

Run: Compile and Run your program. If your program does not compile, fix the syntax errors and compile the program again. Once you are able to successfully run the program, check to see if the Exit button works properly.

Step 3: Double click the main FORM of the program “Form1”. This will create an event handler for the form (e.g., FormLoad) and add the code that initializes the ComboBox and ListBox properties in the event handler:

```
'Initialize the text property of the three Combo Boxes to a NULL string
ComboBoxMake.Text = ""
ComboBoxModel.Text = ""
ComboBoxColor.Text = ""

'Add a number of cars to the "Make" ComboBox.
ComboBoxMake.Items.Add("Honda")
ComboBoxMake.Items.Add("Toyota")
ComboBoxMake.Items.Add("Ford")
ComboBoxMake.Items.Add("Lexus")
```

Run: Compile and Run your code again to see if it runs properly. Click on the top most ComboBox. You should see the above car makes (Honda, Toyota, ..) displayed in the ComboBox. Select one of them and note the result. What happened when you selected one of the items? Now try clicking the other two combo boxes. Do you see any items in the Model and Color Comboboxes?

Step 4: Double click the MAKE ComboBox. This will create an event handler for the MAKE combo box. In the event handler sub procedure type the following code:

```
Dim SelectedItem As String
```

```
SelectedItem = CStr(ComboBoxMake.SelectedItem())  'Return the selected item from the ComboBox
MessageBox.Show(SelectedItem)                     'Display the selected item
```

Run: Run your code again to see if it runs properly and check if the SelectedItem is properly displayed. In other words, are you getting the correct returned value from `ComboBoxMake.SelectedItem()`. If you are not getting the correct value ask me to help you.

Step 5: Once you are confident that the MAKE ComboBox event handler is working correctly, the next step is to properly populate the MODEL ComboBox, based on the value of the SelectedItem from the Make ComboBox. For example if "Honda" is selected by the user, we would like to populate the MODEL ComboBox with Honda models such as (Accord, Civic, etc). Enter the code below in the MAKE ComboBox (right below the line: `MessageBox.Show(SelectedItem)`).

```
If SelectedItem = "Honda" Then
    ComboBoxModel.Items.Add("Accord")
    ComboBoxModel.Items.Add("Civic")
    ComboBoxModel.Items.Add("CRV")
    ComboBoxModel.Items.Add("Pilot")
    ComboBoxModel.Items.Add("Odyssey")

ElseIf SelectedItem = "Toyota" Then
    ComboBoxModel.Items.Add("Camrey")
    ComboBoxModel.Items.Add("Avalon")
    ComboBoxModel.Items.Add("4Runner")

Else
    ComboBoxModel.Items.Add("Not Available")

End If
```

Question? Run the program again. Do you notice any problems? Try selecting Honda as the Make, and then check the Model ComboBox. Does it look ok? Now, try selecting Toyota, then check Model ComboBox. How does it look now? Do you notice a problem?

Step 6: After answering the above question, you have probably noticed that all the Models are being added to the list. So, by the time you select Toyota, the Model ComboBox will hold the combination of all Honda, as well as Toyota models. Obviously, this is a problem. The way to remedy this is to clear (empty) the Model ComboBox before you add the proper Models for each Make. Examine the lines below. The first line will clear the text property of the Model ComboBox, so the user does not see any residue information from prior selections. The second line clears all the Items in the Model ComboBox collection. Enter these two lines before the IF statement in Step 5.

```
ComboBoxModel.Text = ""                 'Initialize the text property to a NULL string
ComboBoxModel.Items.Clear()             'Clear all the items in the Model combobox collection.
```

Run: Run your code again and again to make sure when a Make is selected, the correct Models appear in the ComboBox. (No more, no less!)

Step 7: Once you are confident that the Model ComboBox has the correct information, repeat steps 4 to 6, but this time for the Color ComboBox. (Note that this code should be placed in the MODEL combobox event handler. It is that handler which should load the colors for the next combobox)

For the "Accord" model:

```
ComboBoxColor.Items.Add("White")
ComboBoxColor.Items.Add("Red")
```

For the "Civic" model:

```
ComboBoxColor.Items.Add("Red")
ComboBoxColor.Items.Add("Yellow")
ComboBoxColor.Items.Add("Green")
```

For "Camrey"

```
ComboBoxColor.Items.Add("Black")
ComboBoxColor.Items.Add("Red")
ComboBoxColor.Items.Add("Yellow")
```

Others

```
        ComboBoxColor.Items.Add("Not Available")
```

Run: Run your code again to make sure when a Make is selected, the correct Models appear in the ComboBox. Similarly, when the Model is selected the correct Colors for that model appear in the Color ComboBox. (No more, no less!) If you see any problems try to correct it yourself, if you have tried and still can't find the solution, ask.

Step8: Once all the Combo Boxes work as expected. We will turn our attention to the "Show Selections" button. The purpose of this button is to visually show the result of three ComboBox selections. In other words, we would like to see which Make, Model and Color was selected by the user, and displays that information in the OutputListBox. Create an event handler for the Show Selections button and include the following code in the event handler.

```
ListBoxOutput.ClearSelected()        ' Clear any selections from the ListBox.
ListBoxOutput.Items.Clear()          ' Clear any items that may be in the Output ListBox.


'Get the ComboBox selections and add them to the Output ListBox.
Dim SelectedItem As String

SelectedItem = CStr(ComboBoxMake.SelectedItem())
ListBoxOutput.Items.Add(SelectedItem)       ' Add the Make to the output list box

SelectedItem = CStr(ComboBoxModel.SelectedItem())
ListBoxOutput.Items.Add(SelectedItem)       ' Add the Model to the output list box

SelectedItem = CStr(ComboBoxColor.SelectedItem())
ListBoxOutput.Items.Add(SelectedItem)       ' Add the Color to the output list box
```

Discussion: (Improving our code)

What happens if the user has not selected an item from the above ComboBoxes, yet they click on the Show Selections button? Try this now and note the result. This problem occurs because the programmer has not been careful. He/She is trying to access the SelectedItem() of a ComboBox, before checking to see if such a selection has been made by the user.

In order to remedy this problem, we must make sure the user has selected an item. This can be achieved by checking the SelectedIndex property of the ComboBox. If the SelectedIndex property is something other than a (-1), then the user has selected an item, and we can extract that item from the ComboBox. The highlighted code below demonstrate this point.

```
'Check to see if there are any ComboBox selections, if so add the item to the Output ListBox.
Dim SelectedItem As String

If ComboBoxMake.SelectedIndex <> -1 Then         'A (-1) indicates that no item is selected
    SelectedItem = CStr(ComboBoxMake.SelectedItem())
    ListBoxOutput.Items.Add(SelectedItem)       ' Add the make to the output list box
End If

If ComboBoxModel.SelectedIndex <> -1 Then        'A (-1) indicates that no item is selected
    SelectedItem = CStr(ComboBoxModel.SelectedItem())
    ListBoxOutput.Items.Add(SelectedItem)       ' Add the Model to the output list box
End If

If ComboBoxColor.SelectedIndex <> -1 Then        'A (-1) indicates that no item is selected
    SelectedItem = CStr(ComboBoxColor.SelectedItem())
    ListBoxOutput.Items.Add(SelectedItem)       ' Add the color to the output list box
End If
```