Lab 3 – User Interface Design Using VB.NET

Human Computer Interfacing

(BESE 14)

Objectives

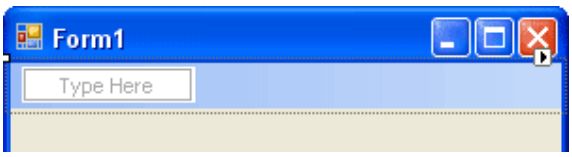
This is the second lab activity on designing a user interface using VB.net. In the previous lab we had seen how to use 'Labels', 'Text boxes' and 'Buttons' to an interface. Today we will focus on:

- Menus
- Picture Box
- Combo Box
- Check Boxes and Radio Buttons
- Calendar

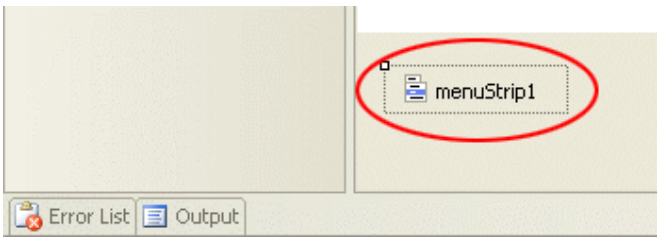
Task 1

Note: This task comprises practice exercises and you do NOT need to submit any lab report/program code.

- Launch your Visual Basic .NET or Visual Studio software and start a new project.
- Add a 'MenuStrip' control to your Form. When you do, you'll notice two things. At the top of your form, you'll see this:

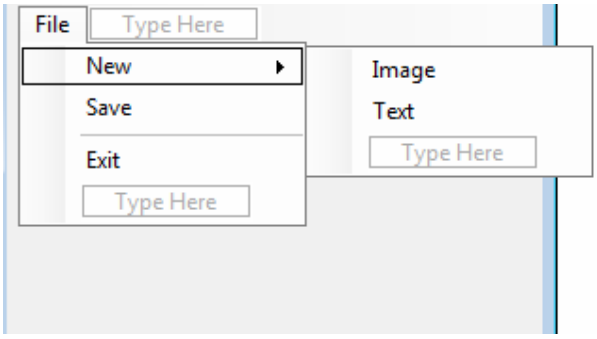


At the bottom of your screen, on the left. You'll see this:



- At the bottom is the control itself. If you click on this, you'll see that the Properties box on the right changes.
- At the top is the ‘MenuItem’ object - The one that says Type Here.
- To start building your menu, click inside the area that says "Type Here". Type the word File and press Enter

- In the same way add menus and get some thing like:



- The final item added to the menu is an "Exit" item. But you can add a separator between the "Save" and "Exit".
- To add a separator, click inside the blue "Type Here" box. Instead of typing a letter, type the minus character "-" . When you hit your return key, you'll see the separator appear.
- Add the following code to the Exit menu click event:

```
Me.Close()
```

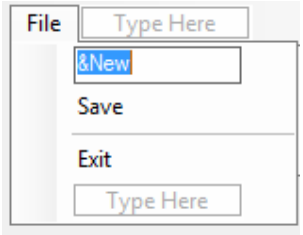
- The word "Me" refers to the form. When you type the word ‘Me’, you'll see a list if items appear. Double click the word ‘Close’, then press your return key.
- To test out your code, run your program. Click your **File** menu, and then click the **Exit** item. Your form should close down, and you'll be returned to the design environment.

Add Shorcuts to your Menu Items

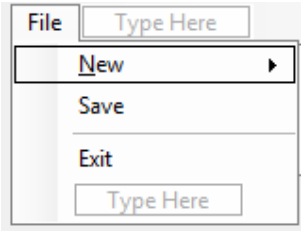
Underline Shortcut

To add an underline short cut, do this:

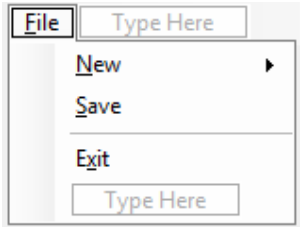
- Click on your New menu item once. This will select it
- Position your cursor before the "N" of New
- Type an ampersand symbol (&)



- Hit the return key on your keyboard
- You should see this:



- Notice that "N" of New is now underlined. If you want an underline shortcut, the ampersand character should be typed before the letter you want underlined.
- Add underlines for the "F" of you File menu, the "S" of Save, and the "X" of Exit. When you're done, your menu should look like this one:



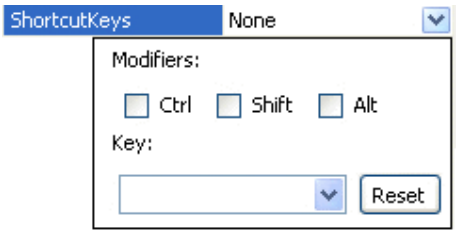
To see how the shortcut works, run your program. To use the underline shortcuts on menus, you first hold down the Alt key on your keyboard. Then type the underline character.

- Hold down the Alt key while your program is running
- Press the letter "F" on your keyboard
- Then press the letter "X" (for the Exit menu)
- Your program should close down

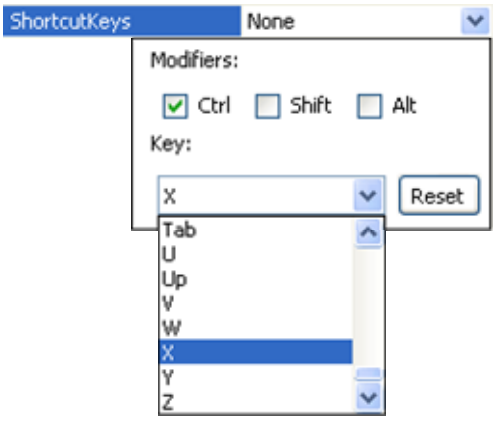
Key combination shortcuts

A key combination shortcut is one that appears at the end of a menu item (Ctrl + X, for example). You can easily add this option to your own programs.

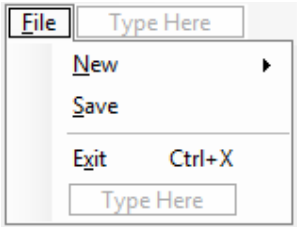
- In Design time, select the Exit item on your menu
- Look at the properties box on the right
- Locate the ShortcutKeys item and click the down arrow to reveal the following:



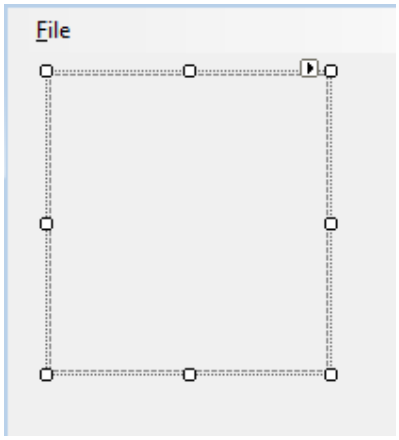
- The Modifier is the key you press with your shortcut. For example, the CTRL key then the "X" key on your keyboard. Place a check inside the Ctrl box. Then select the letter "X" from the Key dropdown list, as in the next image:



- Click back on your menu to see what it looks like:



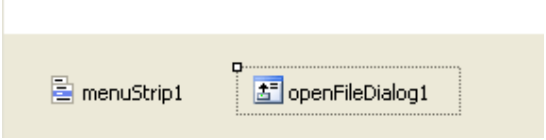
- Run your program and test out the shortcut. Don't click the File menu. Just hold down the Ctrl key on your keyboard. Then press the letter X. Again, the program will close down.
- Now add a “Picture box” to your form to have some thing like this:



- Select the Picture Box and change the Property “SizeMode” to “StretchImage” so that the image we load in the Picture box is stretched to fit in.

Adding the File Open Dialog

- When we click on File > New > Image from our menu, we want the Open File dialogue box to appear allowing us to select an image that is displayed in the Picture Box. This is fairly straightforward in VB.NET. In fact there is even a control for it.
- Open up your toolbox, and locate the control called "OpenFileDialog".
- Double click the control to add one to your project.
- But notice that the control doesn't get added to your form. It gets added to the area at the bottom, next to your menu control:



- Change its Name property to openFD.
- Access the code for your File > New > Image menu item and add the following code:

```
openFD.ShowDialog()
```

- This method of the OpenFileDialog shows the file open dialogue box. You can even test it out right now. Press F5 to run your program. Then click the Open item on your File menu. You should see an Open dialogue box display.
- Return to the design environment, and we'll explore some more things you can do with this Dialogue box control.
- You can set which directory the dialogue box should display when it appears. Instead of it displaying the contents of the "My Documents" folder, for example, you can have it display the contents of any folder. This done with the Initial Directory property. Amend your code to this:

```
openFD.InitialDirectory = "C:\"
openFD.ShowDialog()
```

- By default, the dialogue box will display the word "Open" as a caption at the top of your dialogue box. You can change this with the Title property. Add the line in Bold to your code:

```
openFD.InitialDirectory = "C:\"
openFD.Title = "Open an Image File"
openFD.ShowDialog()
```

- In most dialogue boxes, you can display a list of specific files that can be opened. These are displayed in the "Files of Type" drop down list. To do this in VB.NET, you access the Filter property. We'll restrict our users to only opening JPG files, those that end in the extension ".jpg".

```
openFD.InitialDirectory = "C:\"
openFD.Title = "Open an Image File"
openFD.Filter = "Jpeg Files|*.jpg"
openFD.ShowDialog()
```

- To display files of more than one type, add a Pipe character between each filter. In the code below, two file types are specified:

```
openFD.Filter = "Jpeg Files|*.txt|Gif Files|*.gif"
```

- You'll notice from that if you run the program and select a file and click the Open button, nothing happens. That's because the Open dialogue box doesn't actually open files! It only displays a list of files that CAN be opened. We'll be writing the code that does the opening but for that we need to be able to get the name of the file. The Open Dialogue box has a property that returns the file name that was selected. It's called FileName:

```
OpenFD.FileName
```

- However, this is a property that returns a value (a string value). The value is the name of a file. So you have to assign this value to something. We can assign it to a new variable

```
Dim strFileName As String

openFD.InitialDirectory = "C:\"
openFD.Title = "Open an Image File"
openFD.Filter = "Jpeg Files|*.jpg
openFD.ShowDialog()


strFileName = OpenFD.FileName

MsgBox strFileName
```

- Run your program, and click your File > New > Image menu. Navigate to where you have some jpeg files. Click one to select it. Then click the Open button. You should see the name of the file displayed in a message box.
- Instead of displaying the name of the file in a message box, we would rather like the image to be inserted in our Picture box. To do this we replace the MsgBox with the following:

```
Dim strFileName As String

openFD.InitialDirectory = "C:\"
openFD.Title = "Open an Image File"
openFD.Filter = "Jpeg Files|*.jpg
openFD.ShowDialog()


strFileName = OpenFD.FileName

PictureBox1.Image = Image.FromFile(strFileName)
```

- Now Run your program and try loading an image in the picture box.
- One last thing that we would like to handle is that if the user selects a file and then clicks the Cancel button. You can test to see if it was clicked by assigning the openFD.ShowDialog() to an integer:

```
Dim Result As Integer = openFD.ShowDialog()
```

- You can then test what is inside of the 'Result' variable. If the cancel button is clicked, the result of the action is stored by VB.NET in this property:

```
DialogResult.Cancel
```

- So you may change your code like:

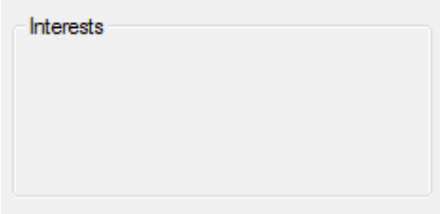
```
If Result = DialogResult.Cancel Then
          MsgBox("Cancel Button Clicked")
Else
        strFileName = openFD.FileName
        PictureBox1.Image = Image.FromFile(strFileName)

End If
```

- Now run and test your program.

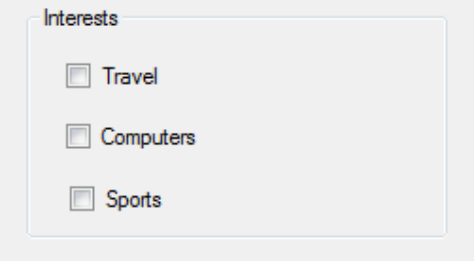
Using Check Boxes

We will now see how check boxes can be added to our interface. Instead of adding check boxes separately, they are generally grouped together, and are moved around as one - by using a Group Box.

- Now locate the Group Box control in the toolbox, draw it on the form and change the Text property to ‘Interests’



- Add a few check boxes to the group:



- If you click on any one of your Checkboxes and examine its Properties in the Property box, you'll notice that it has a CheckState Property. Click the down arrow to see the options this CheckState has. As you can see, you are given three options are Unchecked, Checked, Indeterminate.
- If a checkbox has been selected, the value for the CheckState property will be 1; if it hasn't been selected, the value is zero. (The value for the Indeterminate option is also zero, but we won't be using this.)
- We're only going to test for 0 or 1, Checked or Unchecked. You can do the testing with a simple If Statement. Like this:

```
If CheckBox1.CheckState = 1 Then
```

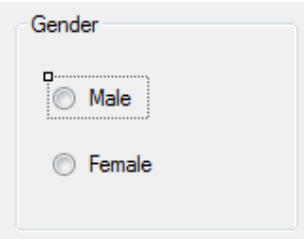
- After you type the equal sign, though, VB will give you a drop down box of the values you can choose from. So the above code is the same as this:

```
If CheckBox1.CheckState = CheckState.Checked Then
```

- Whichever you choose, the Statement will be True if the checkbox is ticked and False if it isn't.

Using Radio Buttons

Radio Buttons, sometimes called Option Buttons, are used when you want to restrict a user's choice to one, Male/Female, for example. A Checkbox would be no good here, because a user could tick both boxes. You want to force your users to pick only one from your list of options.

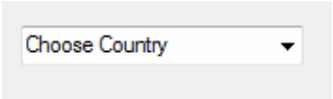


You can check which radio buttons are checked using statements like

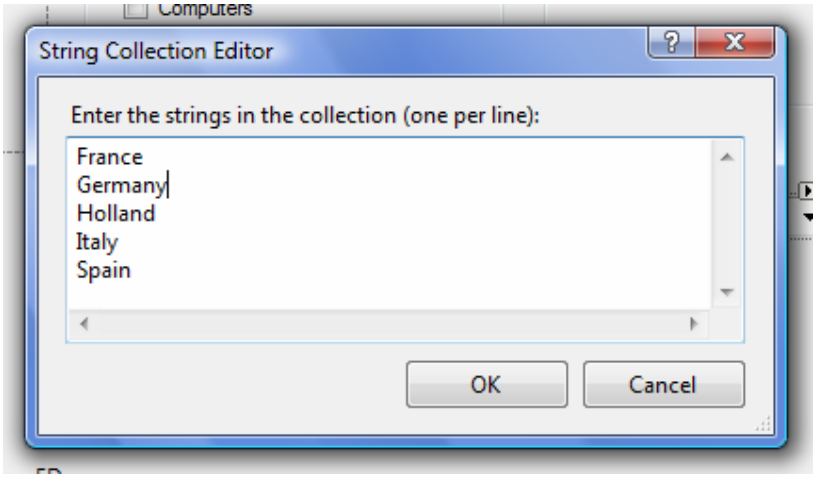
```
Dim selected As String

If RadioButton1.Checked = True Then
selected = RadioButton1.Text

ElseIf RadioButton2.Checked = True Then
selected = RadioButton2.Text
End If
```

Using a Combo Box

Combo boxes are so-named because they "combine" the features found in both text boxes and list boxes. Combo boxes are also commonly referred to as "drop-down boxes" or "drop-down lists". The combo box displays the selected item in the text box portion of the combo box. The ListBox portion of the combo box remains hidden until the combo box receives focus and the user clicks the down arrow on the text box.



- Add a combo box to your form and set the ‘Text’ property to ‘Choose Country’. It will be displayed in the combo box once you execute the program.
- Click on the ‘Items’ Property and add a few Countries to the combo box.

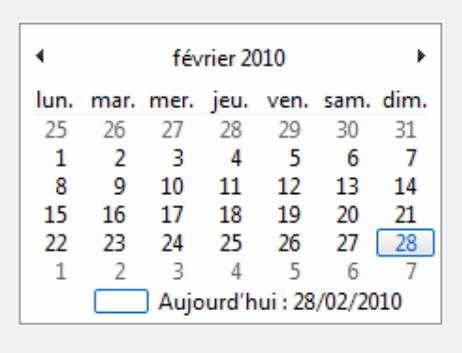


- You may also set the 'Sorted' property of the combo box to 'True' so that the list appears in a sorted form.
- To get the value that is selected, you can use:

```
Selected = ComboBox1.Text
```

Using the MonthCalendar Control

Adding a 'MonthCalendar' control allows selecting a date (or a range of dates).



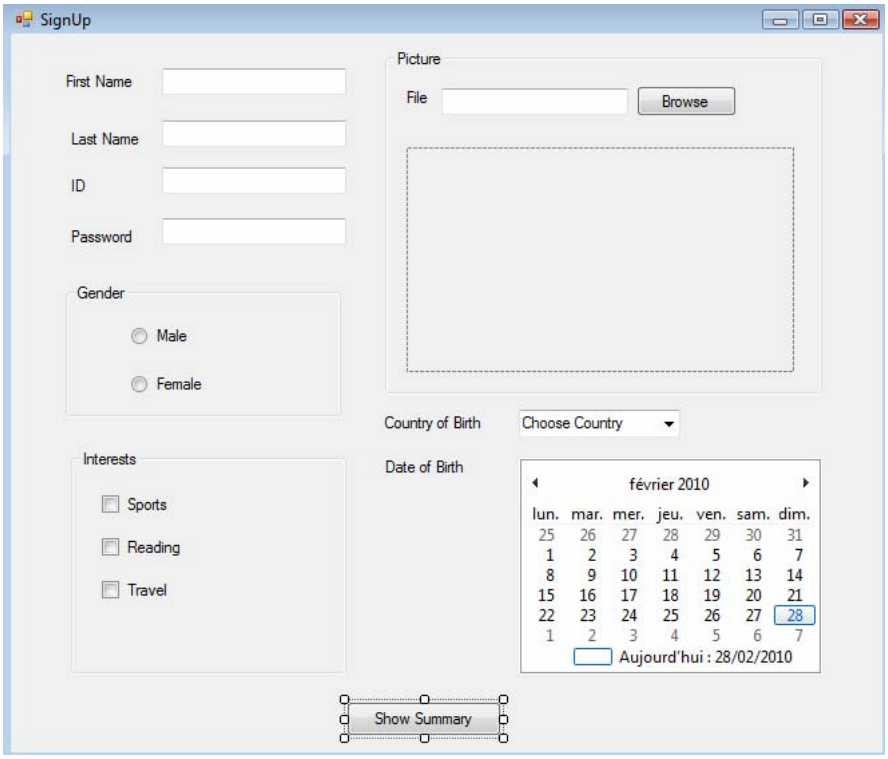
You may get the selected date using:

```
Dim selectedDate As String = MonthCalendar1.SelectionStart
```

Task 2

Note: This task will be considered as an **Assignment**. You should start working on it during the lab session and in case time does not permit, submit it by **SUNDAY, 07 – 03 - 2010**

Using the controls that we have seen, design an interface similar to the following:



Guidelines:

- You may change the layout, Color Scheme or Fonts etc. of the proposed design.
- When you add the “TextBox” for password, you may set the property ‘PasswordChar’ to ‘*’ for example, so that once the user types his/her password the characters are not visible.
- The user may click on the browse button to add an image.
- Once all the information has been filled in, the user clicks the ‘Show Summary’ button to view a summary of all that has been entered.
- Add something like this to the show summary button:

```
Dim summary As String
```

```
summary = "Your first name is " & TextBox1.Text & vbNewLine
summary = summary & "Your last name is " & TextBox2.Text & vbNewLine

 'Use if else statements to add gender and interest summary

summary = summary & "Your date of birth is " &
MonthCalendar1.SelectionStart

MessageBox.Show(summary)
```

You may use Help from the cited reference or any other resource (but do mention it in your code).

Reference:

The practice exercises in Task 1 are based on an online VB.NET tutorial that can be found at:

http://www.homeandlearn.co.uk/NET/vbNet.html