KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 1 : GUI - Overview

By Mohd Taib bin Wahid

Fall 2011-2012

# Lab 1: GUI - Overview

## Objectives

- Understanding of the importance, benefits, and usage of the
  - Memory
  - Physical control and displays
  - WIMP

## Activity Outcomes

- The students will know what are memory, physical control and WIMP from the case study

- Understanding of the importance, benefits, and usage of the
  - Memory
  - Physical control and displays
  - WIMP

**EXERCISE –A**

Is it important that system designer takes account of individual differences when creating an interactive system? Justify your answer. (Group Work)

**SOLUTION – A**

**EXERCISE –B**

From what you have learned about cognitive psychology, devise appropriate guidelines for use by interface designers. You may find it helpful to group these under key headings: for example, Functionality, Visual perception, Memory, etc, although some may overlap such groupings.

**SOLUTION –B**

Functionality:

1. Limit the number of things to be remembered to 7, and preferably to 5. In addition, group things according to function.

**EXERCISE –C**

Identify input and output devices that could benefit users with special needs. (Group Work)

**SOLUTION -C**

| | Device | Type (Input/Output) | Explanation |
|---|---|---|---|
| **1** | | | |
| **2** | | | |
| **3** | | | |
| **4** | | | |
| **5** | | | |
| **6** | | | |
| **7** | | | |
| **8** | | | |

**EXERCISE –D**

Find as many different examples as you can of physical controls and displays.
(a) List them
(b) Try to group them, or classify them

**SOLUTION - D**

**a. List of All Items:**

| | | | |
|---|---|---|---|
| **1** | **Monitor** | **11** | |
| **2** | **Projector** | **12** | |
| **3** | **Mobile Phone Screen** | **13** | |
| **4** | **Projector Controller** | **14** | |
| **5** | **Joystick in Mobile Phone** | **15** | |
| **6** | | **16** | |
| **7** | | **17** | |
| **8** | | **18** | |
| **9** | | **19** | |
| **10** | | **20** | |

**b. Group or classify them:**

| Physical Controls | Displays |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

**EXERCISE –E**

(a) Group the following functions under appropriate headings, assuming that they are to form the basis for a **menu-driven word-processing system** - **the headings you choose will become the menu titles**, with the functions appearing under the appropriate one. You can choose as many or as few menu headings as you wish. You may also alter the wordings of the functions slightly if you wish.

save, save as, new, delete, open mail, send mail, quit, undo, table, glossary, preferences, character style, format paragraph, lay out document, position on page, plain text, bold text, italic text, underline, open file, close file, open copy of file, increase point size, decrease point size, change font, add footnote, cut, copy, paste, clear, repaginate, add page break, insert graphic, insert index entry, print, print preview, page setup, view page, find word, change word, go to, go back, check spelling, view index, see table of contents, count words, renumber pages, repeat edit, show alternative document, help.

**SOLUTION – E**

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 2 - Interface Interaction and Data Gathering

By Mohd Taib bin Wahid

Fall 2011-2012

Objectives

- Understanding of the interaction of different interfaces style along with data gathering.

Activity Outcomes

- The students will know that in Human computer interaction course, each device has different types of interaction
- Students will be aware of the advantages and disadvantages of using different interaction styles
- The students will know how ton improve the customer satisfaction by using questionnaire

- Understanding of the interaction of different interfaces style along with data gathering

EXERCISE -A

Pick a couple of computer input devices that you are aware of (joystick, light pen, touch screen, trackball, eyegaze etc.) and note down how each has different attributes that support certain forms of interaction. You ought to know a little about all of these devices - if you do not, research them.

| Input Devices | Attributes | Interaction |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

**EXERCISE -B**

Describe briefly (e.g. advantages, disadvantages, problems, and solutions) four different interaction styles used to accommodate the dialog between user and computer.

Interaction styles: Command line interface, Menus, Natural language, Question/answer, query dialogue, Form-fills and spreadsheets, WIMP interface, Point-and-click interfaces, and Three-dimensional interfaces.

| Interaction styles | Description |
|---|---|
| | |
| | |
| | |
| | |

**Exercise C**

In data gathering, sometime you need to ask some question regarding the working project from the client, or people related to any profession. For this purpose, in this lab you will design questionnaires. There are two types of questionnaires you will design:

(i) Questionnaire for Course Evaluation
(ii) Questionnaire for Customer Satisfaction

**(i) Questionnaire for Course Evaluation**

There are some points for your consideration in this type of questionnaire:

a. You should include both close type of questions **(YES/NO or RATING type)** , and open questions **(WRITE YOUR COMMENT TYPE)** in the questionnaire.
b. The questions can be cover the following topics and you can add other questions if you want.

**a. Interestingness of the course**
**b. Contents of the course**
**c. Usefulness of the course**
**d. Pace of the course**
**e. Material used in the course**
**f. Purpose of the course**
**g. Best thing about the course**
**h. Worst thing about the course**
**i. Comments about the course**

**For example**
The first point can be converted into the following question:

**Do you find the course interesting?**
The response could be in the form of multiple choices of yes, no, somewhat. Alternatively the question can be asked the following way:

**How interesting was the course?**
In this case the response format can be a scale from 1 to 10.
Similarly convert other points into questions too.

The **LAST THREE POINTS** can be used as an example of **OPEN ENDED QUESTIONS**: For example: **"Describe the thing that you like the best about this course".**

At the top of the page you have to create areas for specifying course name, number etc.

**(ii) Questionnaire for Customer Satisfaction**

Similarly another questionnaire will be designed for customer satisfaction. The points to include are:

- **Price of the Product**
- **Market Availability**
- **Usability**
- **Product Awareness**
- **Offering New Version after sometimes**
- **Who can buy?**
- **Physical Structure**
- **Taste**
- **Other Product having same taste or color**
- **Providing Feedback Materials**

Follow above these points and prepare the questionnaire as discussed in point (i).

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 3 - Identifying Needs and Establishing Requirements

By Mohd Taib bin Wahid

Fall 2011-2012

Objectives

- Understand of the establishing requirements for different systems.
- Understand the importance, benefits, and usage of establish a specific, unambiguous, and clear different type of requirements (functional, data, Environmental, user characteristics, and usability goals).

Activity Outcomes

- The students will know how to establish requirements.
- The students will learn to change textual representation of an HTA description to a diagrammatic form on the hierarchal tasks analysis (HTA).

- Understand of the establishing requirements for different systems
- Understand the importance, benefits, and usage of establish a specific, unambiguous, and clear different type of requirements (functional, data, Environmental, user characteristics, and usability goals).

## EXERCISE -A

Suggest some key requirements in each category (functional, data, Environmental, user characteristics, and usability goals) for each of the following systems. In addition, what factors (environmental, user, usability) would affect the following systems most? (Where 1=Most important and 3= Least important)

- Self-service cafeteria in a university's - paying using credit system

| | |
|---|---|
| Functional | Calculate total cost of purchases |
| Data | Access to price of products in cafeteria |
| Environmental | Noisy and busy environment, users maybe talking while using the system |
| User characteristics | Majority of users under 25, comfortable dealing with technology |
| Usability goals | The system should be easy to learn so that new users can use it immediately, and memorable for frequent users. Users would not want to wait around for the system to finish processing, so system needs to be efficient |
| Factors | 1. Environmental 2. Usability 3. User characteristics |

- Automated Teller Machine (ATM) - Getting cash

| | |
|---|---|
| Functional | |
| Data | |
| Environmental | |
| User characteristics | |
| Usability goals | |
| Factors | 1. 2. 3. |

- Fashion clothes website – Search for clothes and add items to basket

| | |
|---|---|
| Functional | |
| Data | |
| Environmental | |
| User characteristics | |
| Usability goals | |
| Factors | 1. 2. 3. |

## EXERCISE –B

Figure 1 shows a textual representation of an HTA description of sending email message to your teacher. Present the same information in a diagrammatic form

### Textual HTA

**Hierarchy description**

0. Send assignment 1 to your teacher via your email
1. Access your email website
2. Login to your email
    - 2.1 Click "Sign in" button or link
    - 2.2 Type in your user name and password
    - 2.3 Press "Sign in" button
3. Click on the "New" link
4. Write your email message and attach file
    - 4.1 Specify the email address of the recipient in "To"
    - 4.2 Write the email subject in the "Subject"
    - 4.3 Type the main email message in the "Message content" box
    - 4.4 Attaching files with your email
        - 4.4.1 Click the "Attach" link
        - 4.4.2 Upload the "Assignment 1" from your computer
5. Send it to the recipient by clicking on the "Send" link
6. Logout Account
    - 6.1 Click "Logout" button
7. Close Web Browser

**Plans**

Plan 0: do 1-2-3-4-5 optional 6-7
Plan 2: do 2.1 – 2.2 – 2.3 if username and password is incorrect do 2.2 - 2.3
Plan 4: do 4.1 – 4.2 – 4.3 – 4.4
Plan 4.4: do 4.4.1 – 4.4.2

Figure 1: An HTA for sending "Assignment 1" to your teacher via your email

Diagrammatic HTA

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 4 - User Interface Development using Visual Basic – Part I

By Mohd Taib bin Wahid

Fall 2011-2012

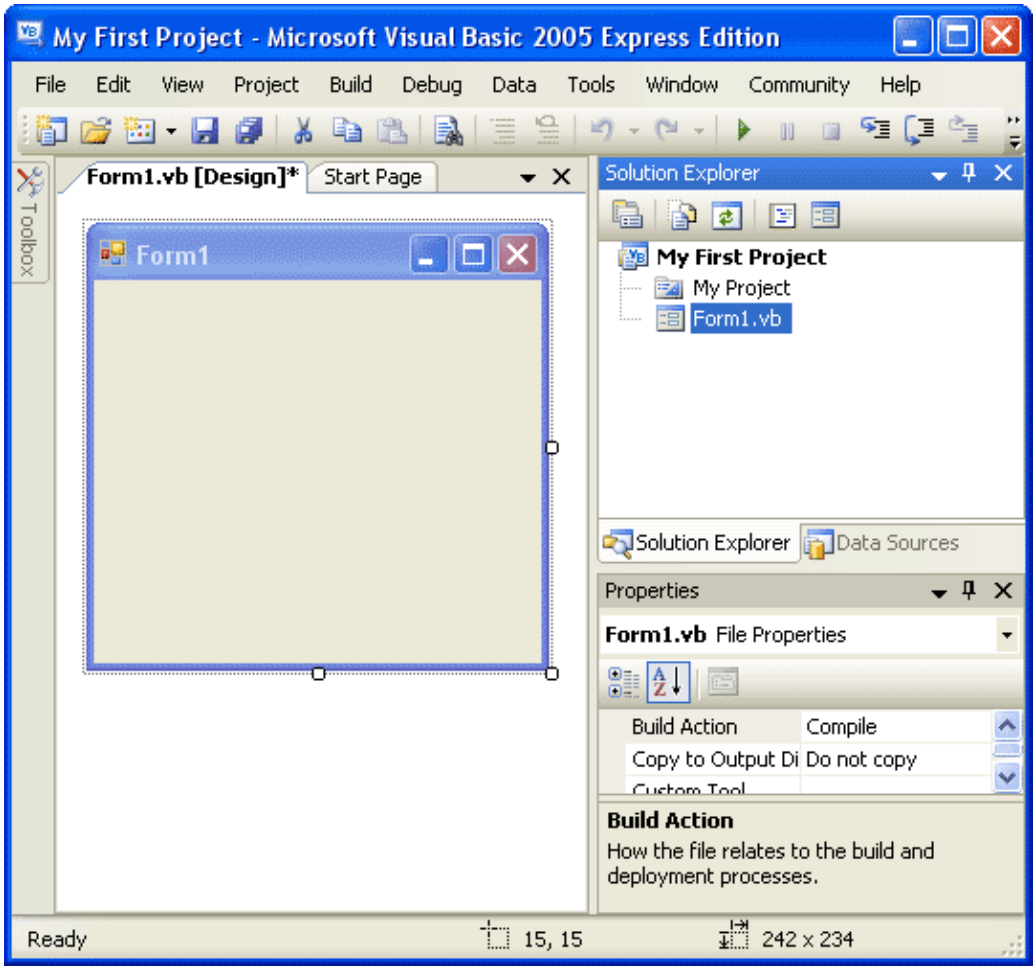
Objectives

- Understanding of the user interface development using Visual Basic

Activity Outcomes

- The students will know how to develop dynamic interface by using visual basic

**My First Project**



- Understanding of the user interface development using Visual Basic

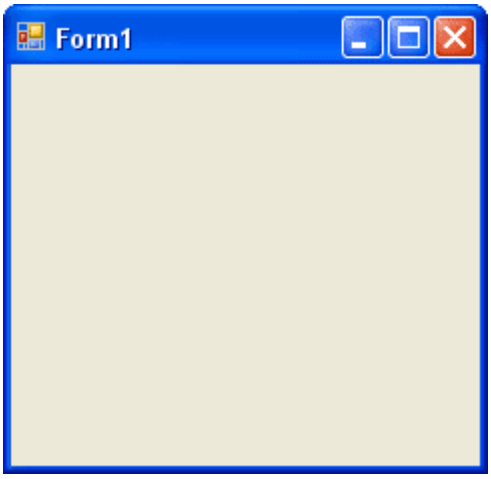
**Visual Basic .NET Forms**

# The Default Form

To run the form, try this:

- From the menu bar, click **Debug**
- From the drop down menu, click **Start**
- Alternatively, press the **F5** key on your keyboard
- Your programme is launched

Congratulations! You have now created your very first programme. It should look like this:




Click the Red X on the form to stop it from running. You will then be returned to the software environment.

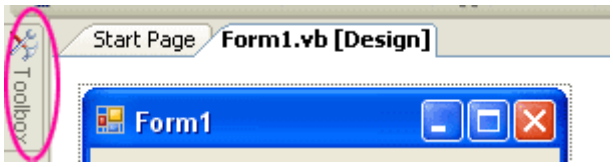
Well, Visual Basic has two distinct environments, a **Design** environment and a **Debug** environment. Design Time is where you get to play about with the form, spruce it up, add textboxes, and buttons, and labels (and code, of course ); Debug is where you can test your programme and see how well it performs.

But don't worry about the terminology. Just be aware that there's a two step process to VB programming: **designing** and **debugging**.

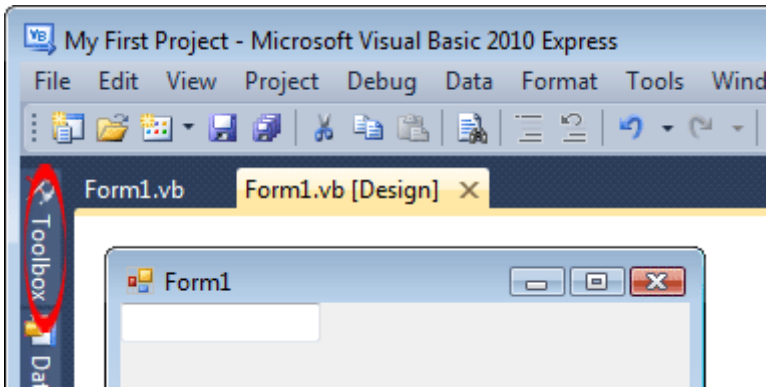
# Adding a Control to a Form

Buttons, textboxes, and labels are all things that you can add to your Forms. They are known as Controls, and are kept in the Toolbox for ease of use.

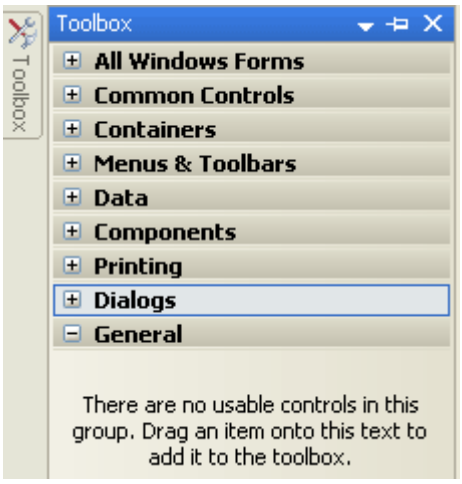
The Toolbox can be found on the left of the screen. In the picture below, you can see the toolbox icon next to Form1 (VB Net 2008):

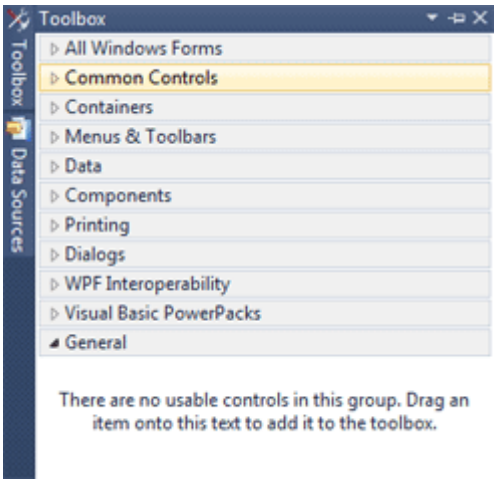



VB Net 2010:

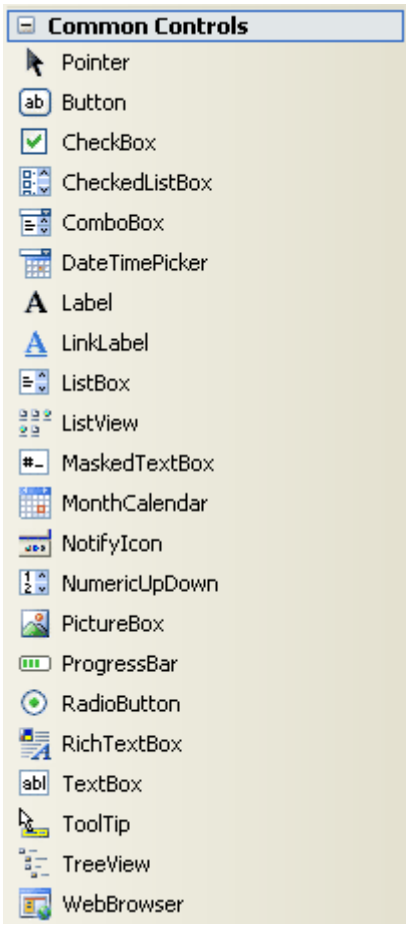



To display all the tools, move your mouse over the toolbox icon. You'll see the following automatically appear:




There are seven categories of tools available. The toolbox you'll be working with first is the Common Controls toolbox. To see the tools, click on the plus symbol next to **Common Controls**. You'll see a long list of tools:




To keep the toolbox displayed, click the Pin icon next to the X. To close the toolbox, simply move mouse away.

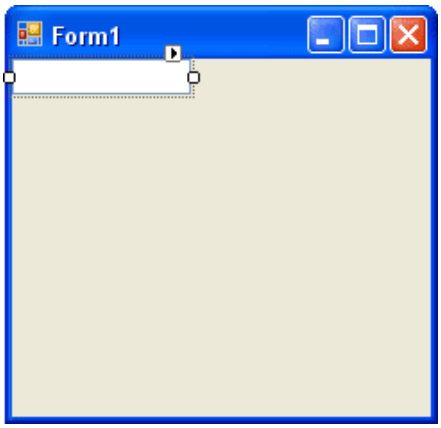
As you can see, there are an awful lot of tools to choose from! For this first section, we'll only be using the **Button**, the **TextBox** and the **Label**.

# How to Add a Control to VB .NET Forms

Let's start by adding a textbox to our form. With the tools displayed, do the following:

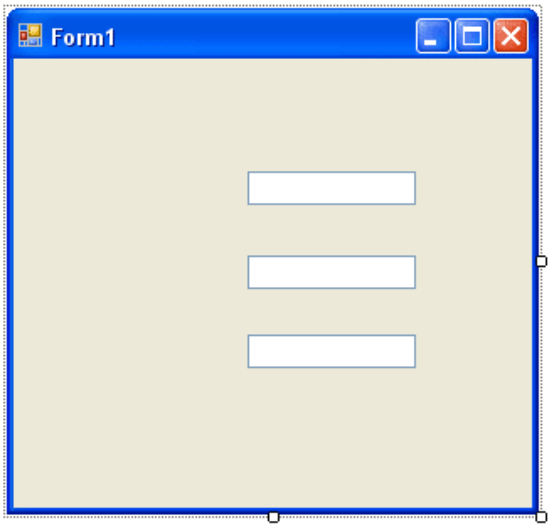
- Locate the TextBox tool
- Double click the icon
- A textbox is added to your form

The textbox gets added to the top left position of your form.




Notice the small squares around the textbox. These are sizing handles. Move your mouse over one of them. You can't make the size any higher, but you can make it wider, because the default action of a textbox is to have it contain only a single line of text. A textbox can only be made higher if it's set to contain multiple lines of text.

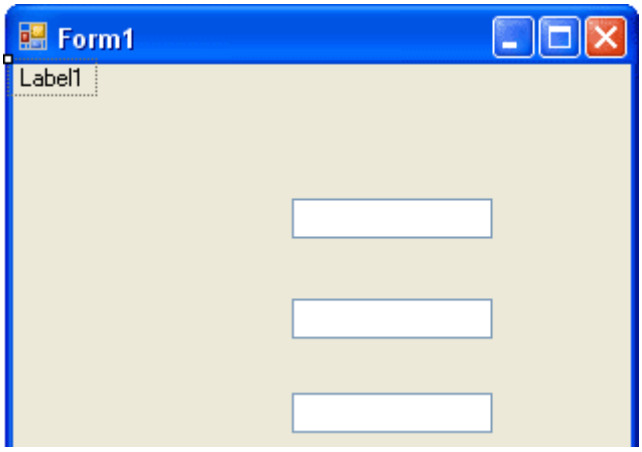
- Create two more textboxes by double clicking on the textbox icon in the toolbar (Or Right-click on the selected textbox and choose Copy. Then Right-click on the Form and choose Paste.)
- Resize them to the same size as your first one
- Line them up one below the other with space in between
- Try to create something that looks like the one below



### Adding a Label to your Form

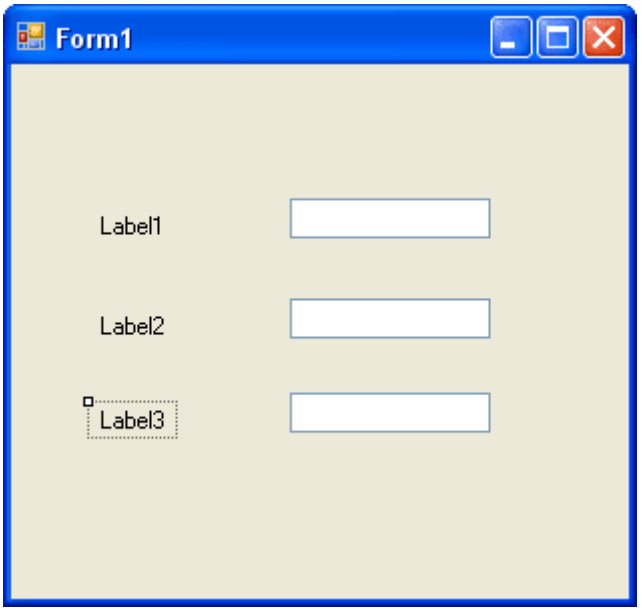
Let's add some labels near the textboxes so that your users will know what they are for.

- Locate the label control in the toolbox
- Double click the label icon
- A new label is added to your form
- It should look like the one below

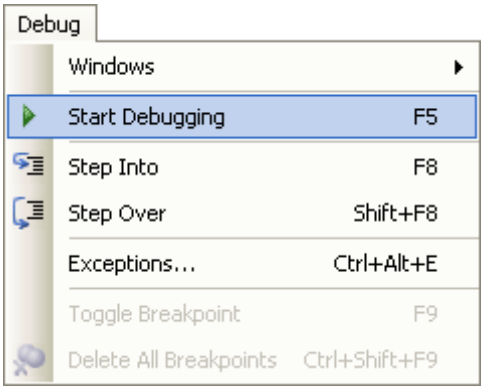



Click on the label to select it. Now hold your left mouse button down on the label. Keep it held down and drag it to the left of the textbox.

Create two more labels, and position them to the left of the textboxes like this one:

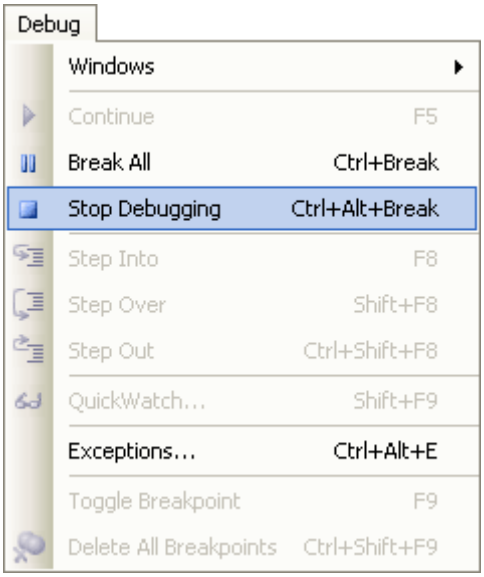



To see what your Form looks like as a programme, click **Debug > Start** from the menu bar. Or press F5 on your keyboard:




To stop the programme from running, you can do one of the following:

1. Click the Red X at the top right of your Form
2. Click **Debug > Stop Debugging** from the menu bar
3. Press Shift + F5 on your keyboard



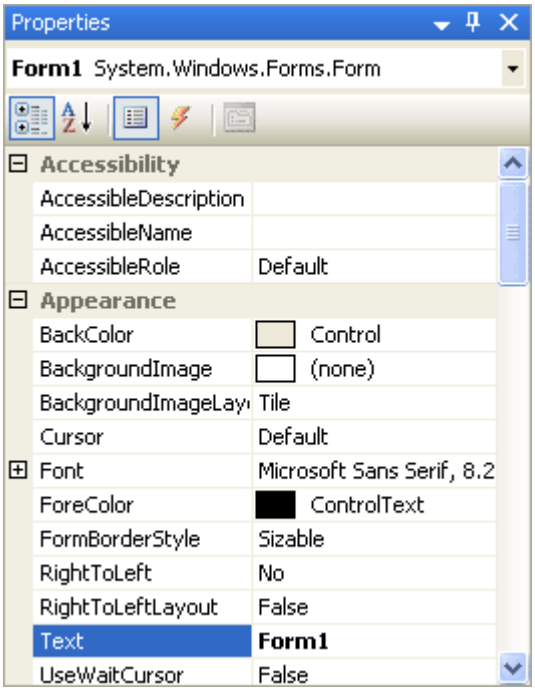

You can also click the Stop button on the VB toolbars at the top, as in the image below:

We now have a form with textboxes and labels, something that looks like a form people can fill in.

# An Introduction to VB .NET Properties

Click anywhere on the form that is not a label or a textbox, somewhere on the form's grey areas. The form should have the little sizing handles now, indicating that the form is selected.

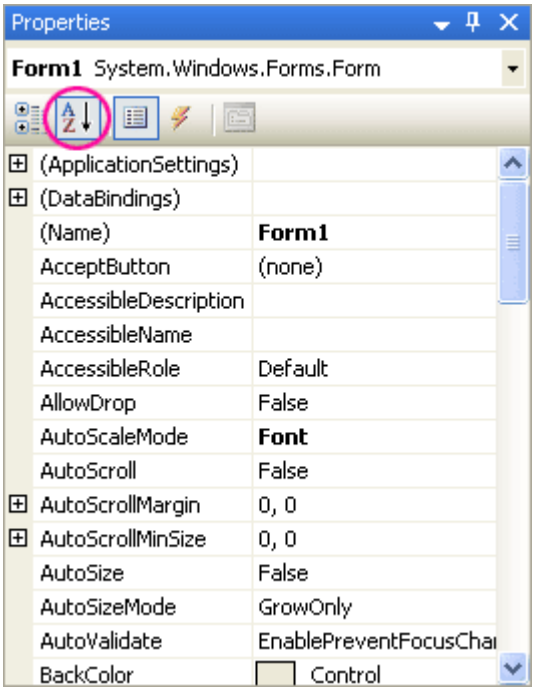
On the right of the design environment there should be the following Properties box:




If Properties box says "Textbox1 Textbox" or "Label1 Label" then you haven't yet selected the Form. Click away from the textbox or label until the Properties box reads "Form1 Form"

What you are looking at is a list of the properties. The rights of these properties are the default values, and can be changed.

To display the list of Properties in an alphabetically. Click the Alphabetic icon at the top of the Properties box, as in the image below:




This will make the properties easier to find.

Before change any in the Properties box, let's clear up what we mean by "Property".

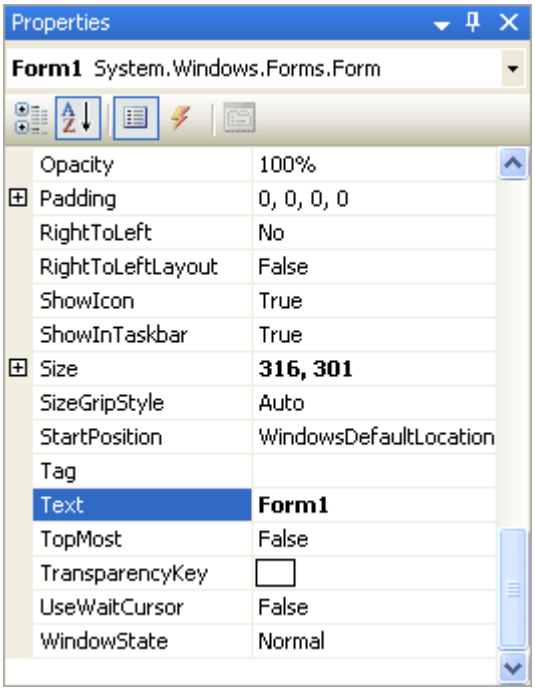
## What is a Property?

Those controls you added to the form (textboxes and labels), and the form itself, are called control objects. You can think of controls as things, something solid that you can pick up and move about. Controls (things) have properties. If your television were a control, it too would have properties: an On/Off button property, a colour property, a volume property, and a ... well, what other properties would your television have? Think about it.

The properties of your television will have values. The On/Off button would have just two values - On or Off. The volume property could have a range of values, from zero to ten, for example. If the value of the volume property was set to ten, the loudest value, then you'd probably have some very angry neighbours!

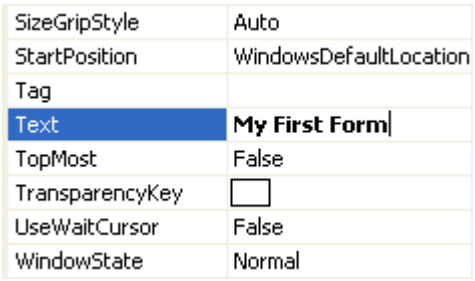
In VB.NET, you can change a property of a control from the Properties Box. (You can also change a property using code, which you'll do quite a lot.) If we go back to our Form object, and the properties and values it has, we can see how to change them using the Properties Box. We'll change only one of these values for now - the value of the Text property . So, do this:

- Locate the word "Text" in the Property box, as in the image below




"Text" is a Property of Form1. The default value of the Text property is set to the word "Form1". To change this to something of your own, do this:

- Click inside the area next to "Text", and delete the word "Form1" by hitting the backspace key on your keyboard
- When "Form1" has been deleted, type the words "My First Form"




- Click back on the form itself (the one with the labels and textboxes), or hit the return key on your keyboard
- The words "My First Form" will appear as white text on a blue background at the top of the form

When you've correctly changed the Text property, your Form will then look like this one:

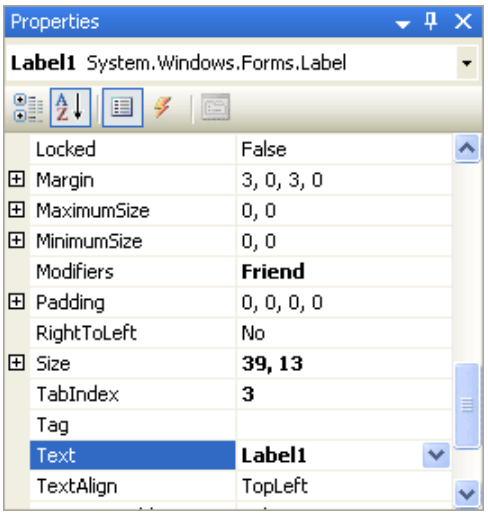

As you can see, your new text has gone at the top of the form, in white on the blue background.

So the Text Property of a form is for setting the caption you want to display in the title bar at the top.

# The Text Property of a Control

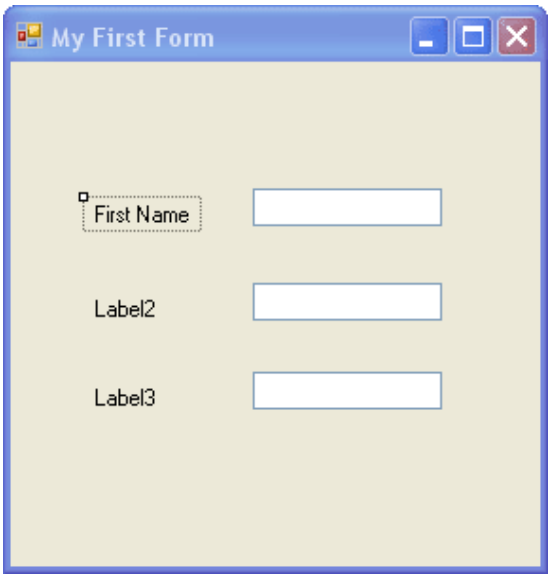
Changing the values of some properties is fairly easy. We'll now change the Text properties of our labels, and the Text properties of our Textboxes.

Click on Label1 so that it has the sizing handles, and is therefore selected. Examine the Property box for the Label:




The Text property of a Label does what you'd expect it to do: adds text to your label.

- With label1 selected, click inside the area next to "Text", and delete the word "Label1" by hitting the backspace key on your keyboard
- Type in the words "First Name"
- Click back onto the grey form, or hit the return key on your keyboard
- Label1 has now changed its text caption to read "First Name"
- If you've made a typing error, go back to the first step above and try again
- Your form should now look like this:

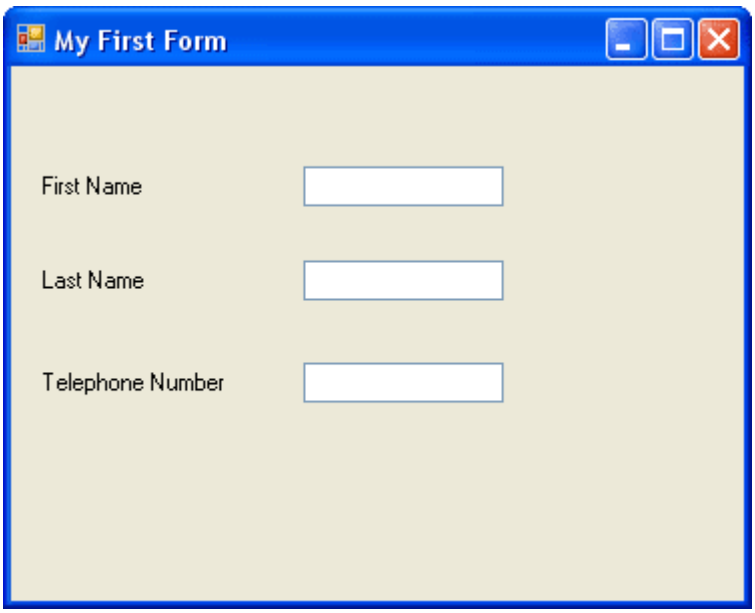


Now, change the Text property of the other two labels. Change them to these values:

**Label2:** Last Name
**Label3:** Telephone Number

The form should look like as below:


The Form can be resized just like the Label and the textboxes. Click anywhere on the form that is not a textbox or a label to reposition and resize the textboxes and labels. Your form might look like this one:



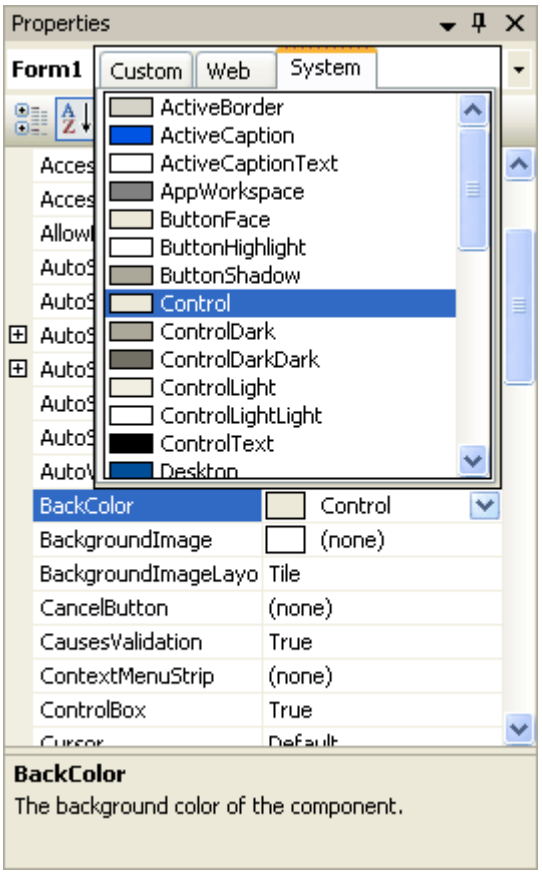

Click on **Debug > Start** to have a look at your programme. Or Press F5 on your keyboard. Click **Debug > Stop Debugging** to get back to the design environment. (Or press Shift + F5, or just click the red X at the top right of the form.)

# Liven up your VB .NET Forms

Changing the colour of the Form means we have to change the BackColor property.

Click anywhere on the form that is not a textbox or a label. The Property Box on the right will read "Form1".

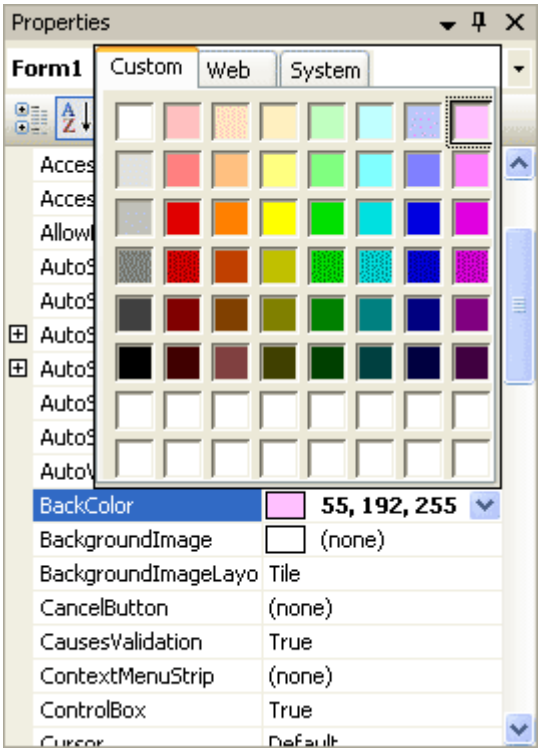
To change the colour of the Form, click the word "BackColor" in the Property Box. Next, click the black down-pointing arrow to the right. A drop-down box will appear.




The default colour is the one selected - **Control**. This is on the **System** Tab. The System colours are to set whatever colour scheme the user has opted for when setting up their computers.

You can choose the colour of the Active Caption. The Active Caption is the one you set earlier when you changed the text to "My First Form".

If you want to choose a colour that is not a System colour, click the Custom Tab. You'll then see this:




Click on any of the Colours in the colour palette and the background colour of your form will change.

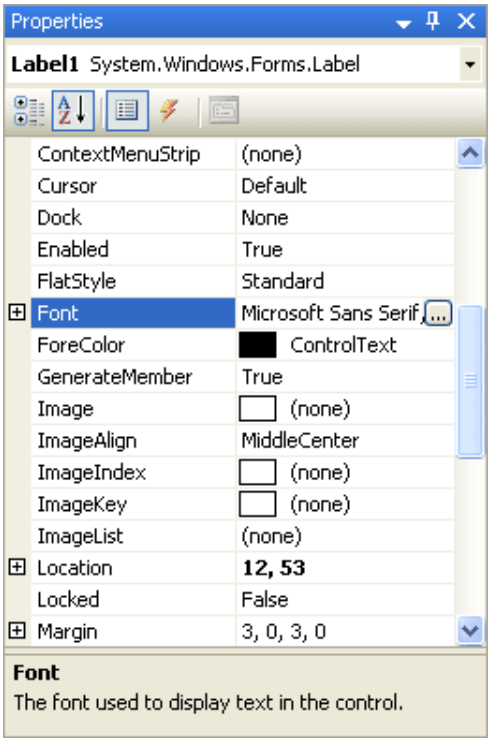
You can also select the Web Tab. When you do, you'll see a list of Web-Safe colours to choose from if you're designing a project for the internet. But you can choose one even if you're not.

To change the colour of the labels, click on a label to select it. Look in the Property box to see if it reads Label. If so, you can now go ahead and change the BackColor property of the Label in exactly the same way that we changed the BackColor property for our Form.

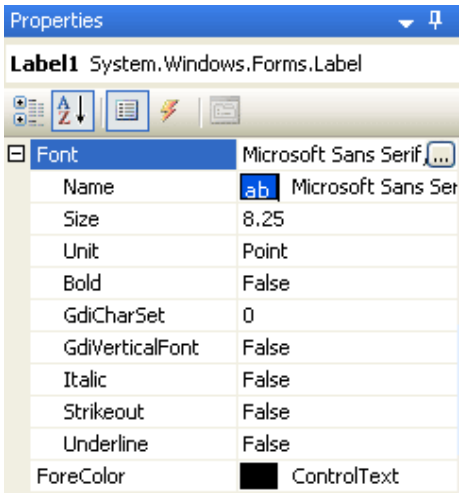
To change the Font size of the Labels and Textboxes, select a control. Let's start with Label1.

- So click on Label 1
- Scroll down the Property Box until you see Font
- Click on the word "Font" to highlight it
- MS Sans Serif is the default Font

Notice that the Font property has a cross next to it. This indicates that the property is expandable. Click the cross to see the following:




Notice that the Font property has a cross next to it. This indicates that the property is expandable. Click the cross to see the following:




You can change a lot of Font properties: the Name of the font, its Size, whether is should be Bold or not, etc. You can also click the square box with the three dots in it to change the font properties in the same place.

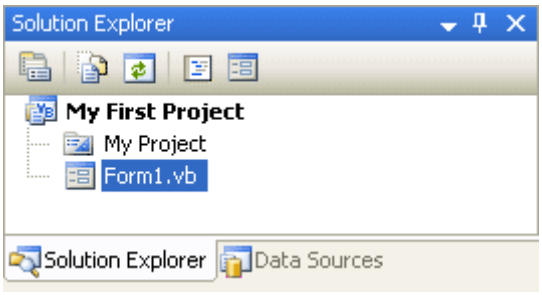
Make the following changes to the three labels:

**Font**: Arial
**Font Style**: Bold
**Font Size**: 10

Change the Font of the three Textboxes so that they are the same as the Labels.

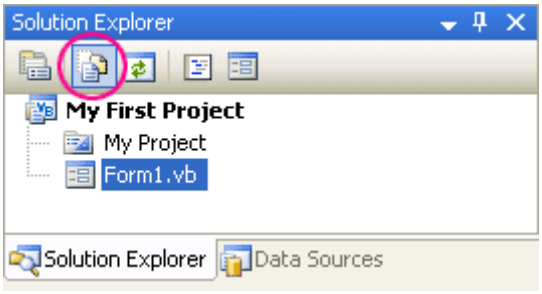
# How to Save your VB .NET Projects

If you have a look in the top right of the Design Environment, you'll see the Solution Explorer. (If you can't see it, click **View > Solution Explore**r in version 2008 and **View > Other Windows > Solution Explorer** in version 2010.)

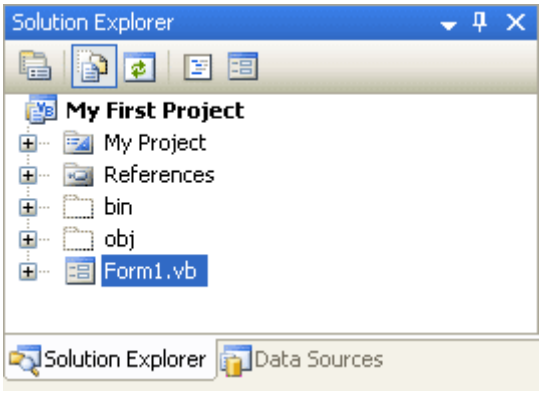



The Solution Explorer shows you all the files you have in your project (Notice that the name of your project is at the top of the tree - "My First Project").

At first glance, it looks as though there are not many files in the project. But click the Show All Files icon, circled below:

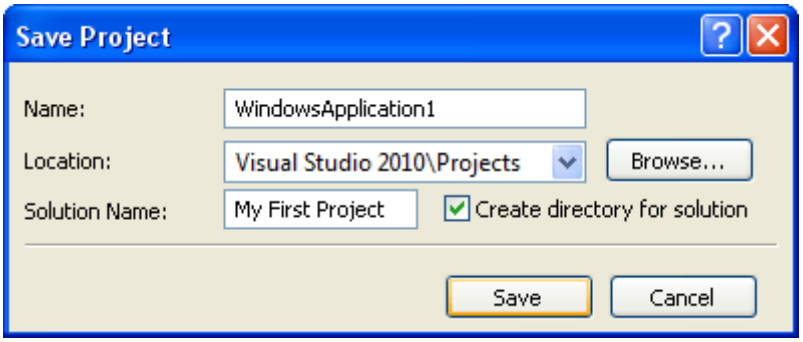



When you click Show All Files, the Solution Explorer will look something like this:




When you save your project, you are saving all these files.

To save your work, click **File > Save All** and you'll see the following dialogue box (we've chopped ours down a bit):




The files are usually saved in the My Document folder in XP (Document folder in Vista and Windows 7), under Visual Studio. If you want to save your projects elsewhere, click the Browse button.

To actually save your work as you go along, just click File > Save All from the menu bar. Or press Ctrl + Shift + S on your keyboard. Or click the icon in the Toolbar (the stack of floppy disks).

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 4 User Interface Development using Visual Basic – Part II

By Mohd Taib bin Wahid

Fall 2011-2012

Objectives

- Understanding of the user interface development using Visual Basic

Activity Outcomes

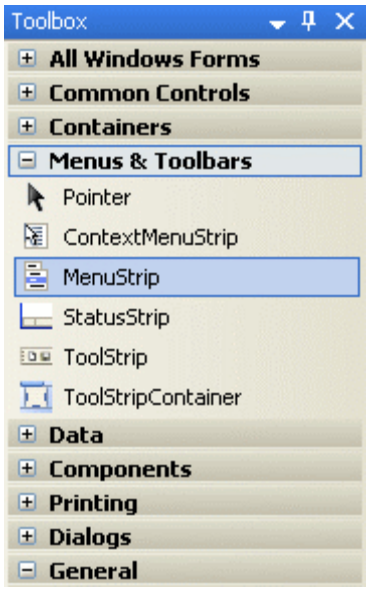
- The students will know how to develop dynamic interface by using visual basic

# Understanding of the user interface development using Visual Basic

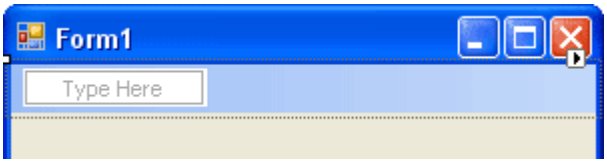
## Adding Menus to a Visual Basic .NET Form

In this section we'll see how to add menus to a Visual Basic .NET form. These types of menus are very common to Windows programme. Visual Basic itself has many of these drops down menus - File, Edit, View, Project, Format, etc. And they're very easy to add.

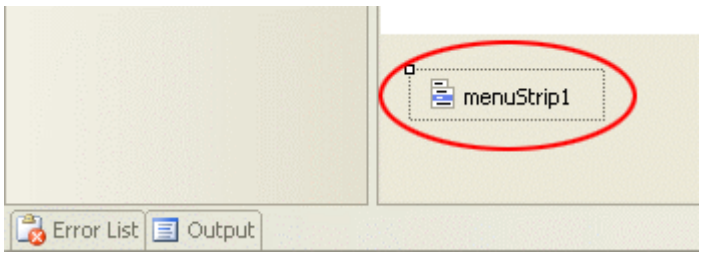
Start a new project. To your new form, use the toolbox to add a MenuStrip control:




Double click the control to add one to your form. When you do, you'll notice two things. At the top of your form, you'll see this:



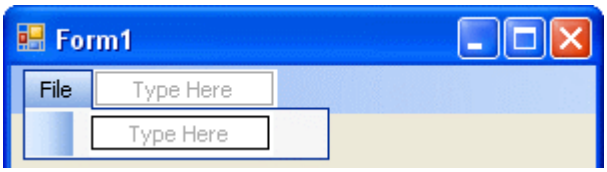

We'll see how to construct our menu soon. But notice the other things that gets added to your project. Examine the bottom of your screen, on the left. You'll see this:



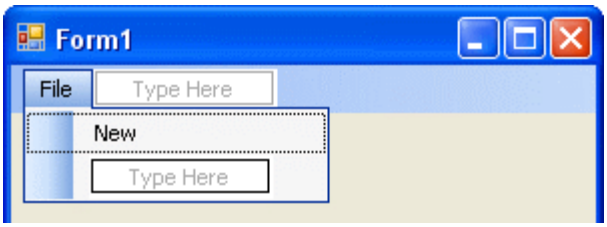
This is the control itself. If you click on this (it's highlighted above), you'll see that the Properties box on the right changes. There are many properties for the control. But there are lots of properties for the MenuItem object. The MenuItem object is the one at the top of the form

To start building your menu, click inside the area that says "Type Here". Type the word File:

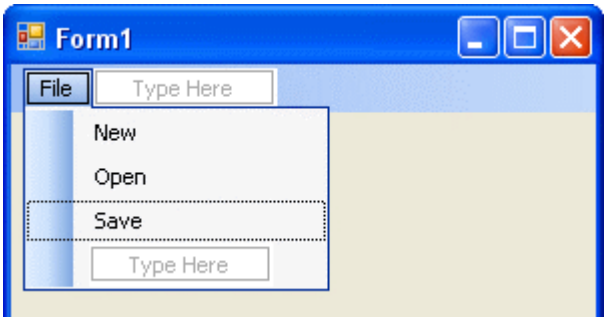

Now press the enter key on your keyboard. Your menu will look like this:




To create items on your File menu, click inside the **Type Here** box. Enter the word **New**, and press the enter key on your keyboard again. Your menu will then look like this:

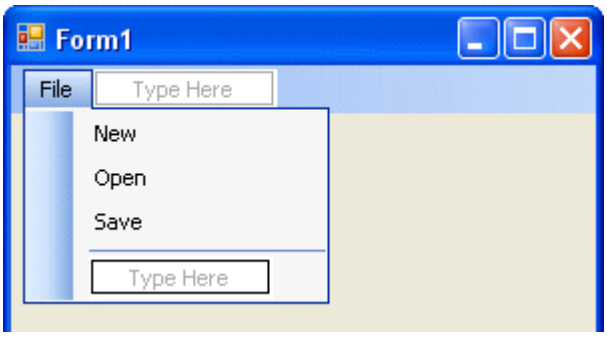



Add an "Open" and a "Save" item to your menu in the same way. It should look like this:




The final item we'll add to our 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 "-" (in between the "0" key and the "+/=" key on your keyboard). When you hit your return key, you'll see the separator appear:




Click inside the "Type Here" area, and add an Exit (or Quit) item. Click back on your form, away from the menu, to finish off. You should now have a File menu like this one:

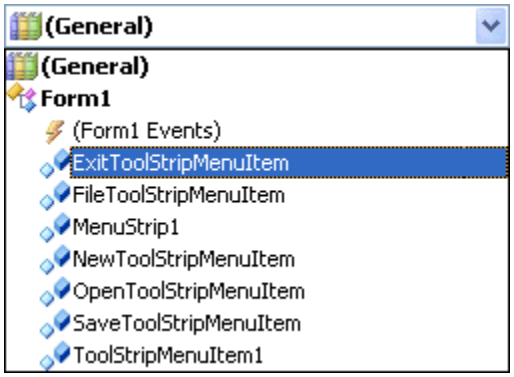

To see what your menu look like, Run your programme. Click the File menu.

## Adding code to a VB.NET menu

Stop your programme and return to the design environment. Click **File** in Design Time to see your drop down menu.

Another way to get to the code for an object is this:

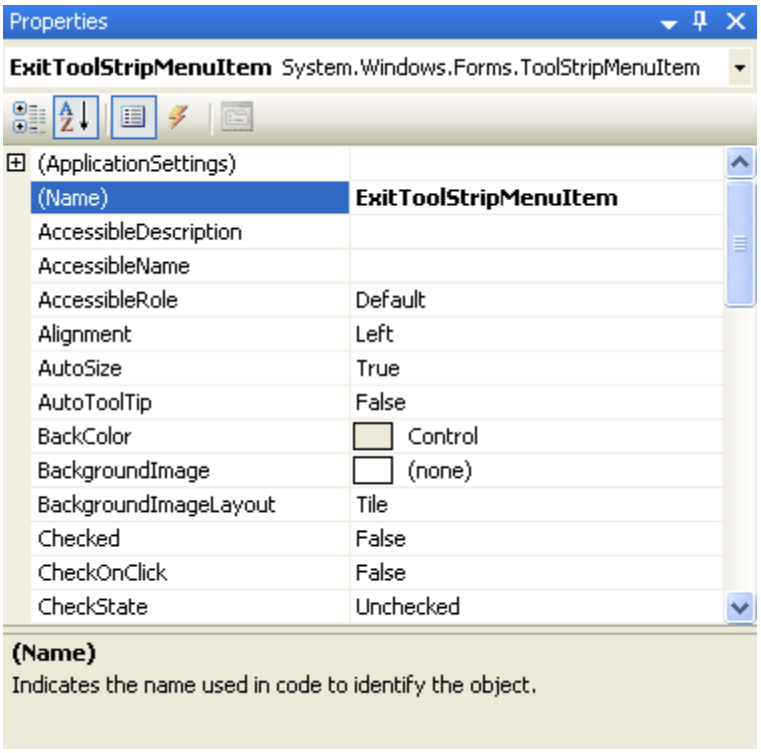
- Press F7 on your keyboard to go to the code window
- Click the black arrow at the top, where it says **General**:




The Exit menu here is "ExitToolStripMenuItem". If you were to click that item, a code stub would open, ready for you to type your code.

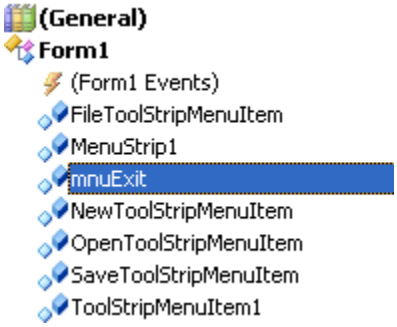
However, "ExitToolStripMenuItem" is very difficult to remember. We can rename our menu items so that they are more descriptive. So do this:

- Get back to your form by pressing Shift + F7 on your keyboard
- Click the **File** menu to select it
- Select your **Exit** (or your Quit) item (Careful not to click in the middle as this may open the code window.Click near the left edge somewhere.)
- When you have the Exit item selected, look at the properties box on the right:

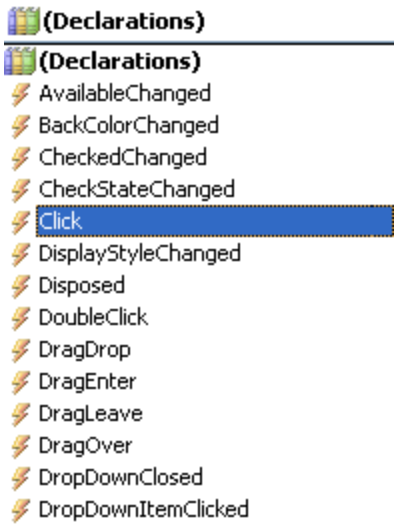



- Click inside the **Name** property
- Change it to **mnuExit** (or mnuQuit)
- Press your return key on your keyboard to confirm the change

Now press F7 again to bring the code window up. Click the drop down arrow of the General box, and you should see the new name appear (Notice that MenuItem6 has vanished):




To jump straight to the code, you need to look at the drop down box opposite. It will probably say "Declarations". Click the arrow and you'll see a new list:




The items in the Declarations box are called Events. The Event you want is the **Click** event. When you select Click from the list, you are taken straight into the code for that event. It should be like this one:

```
Private Sub mnuExit_Click(ByVal sender As Object, _
                          ByVal e As System.EventArgs) _
                          Handles mnuExit.Click

End Sub
```

The code above has been tidied up to fit on this page; yours will all be on one line. But notice that it says **mnuExit_Click**.

What we want to do is add some of our own code, so that out Exit menu item actually does something.

There's only one line of code to add. It's this:

**Me.Close( )**

The word "Me" refers to the form. When your type the word Me, you'll see a list if items appear. Double click the word Close, then press your return key. Your code window should look like this:

```
Private Sub mnuExit_Click(ByVal sender As Object, _
                          ByVal e As System.EventArgs) _
                          Handles mnuExit.Click

    Me.Close()

End Sub
```

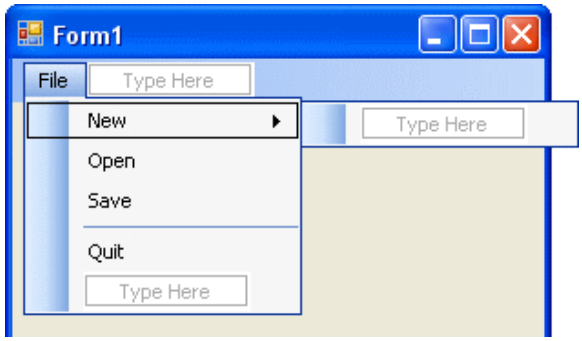
To test out the new code, run your programme. Click your **File** menu, and then click the **Exit** item. Your form should close down, and you'll be returned to the design environment.

## Add a Sub Menu to your VB.NET Form

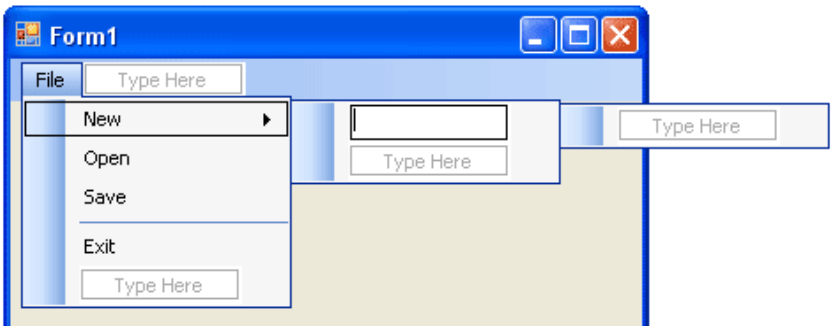
A sub menu is one that branches of a menu item. They usually have an arrow to indicate that there's an extra menu available.

You can create our own sub menus quite easily. Try this:

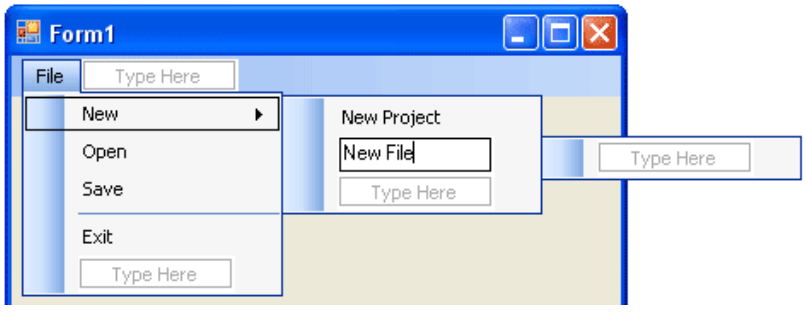
- Return to the Form view (Shift + F7 is a shortcut)
- Click on your File menu so that you can see it all
- Select the New item (Careful where you click. Click once on the left edge). You should see this:



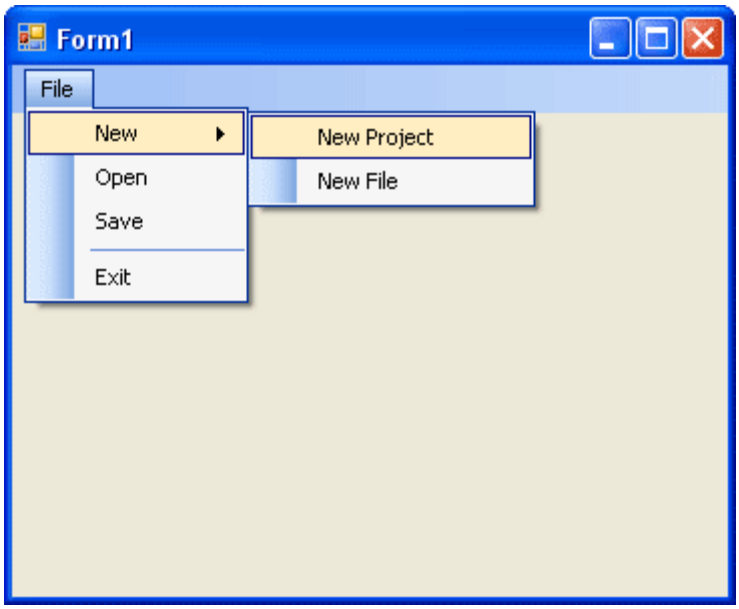

- Click on the "Type Here" just to the right of New
- You'll see yet more "Type Here" areas:




- Type **New Project**, and then hit the return key on your keyboard
- Type in **New File** and then click away from the menu, somewhere on the form
- You will then have a menu like this one:




- Save your work, and then run your programme. Click your new menu to see the following:



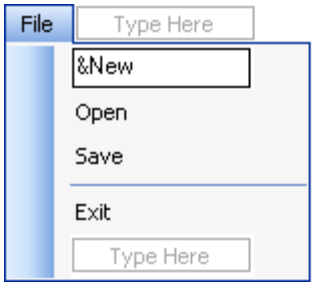

One more thing we can do. If you look closely at a lot of menu items, you see that they have shortcuts attached. There's two types of shortcuts: An underline shortcut, and a key combination shortcut.

## Add Shorcuts to your Menu Items

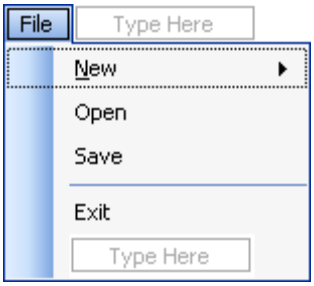
### Underline Shortcut

To add an underline, 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 (**&**)

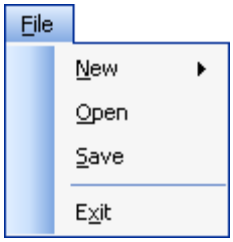



- Hit the return key on your keyboard
- You should see this:




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 "O" of **Open**, the "S" of **Save**, and the "X" of **Exit**. When you're done, your menu should look like this one:



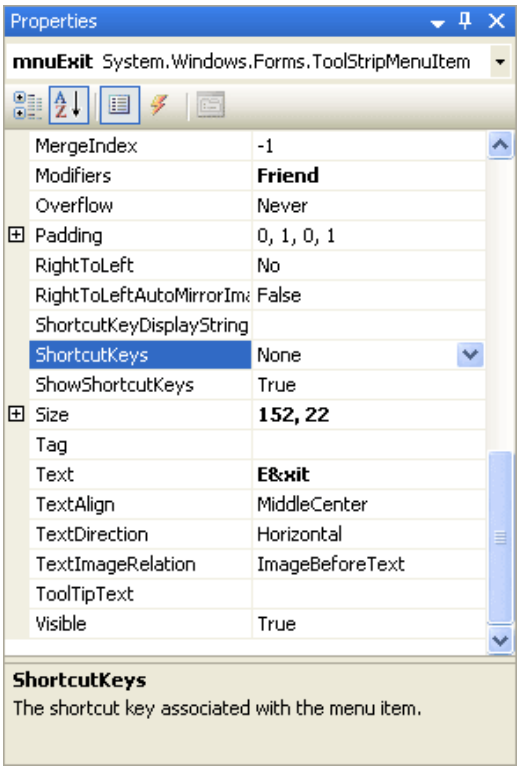
To see your shortcut works. Run your programme. To use the underline shortcuts on menus, you first hold down the Alt key on your keyboard. Then type the underline character.

- Hold down the Alt key while your programme is running (You might not be able to see the underlines until you press the Alt key.)
- Press the letter "F" on your keyboard
- Then press the letter "X" (for the Exit menu)
- Your programme should close down

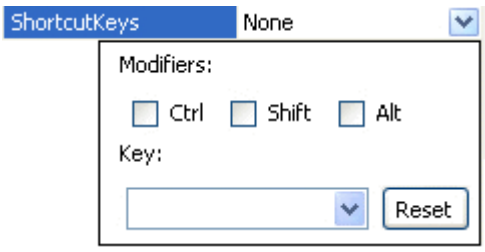
**Key combination shortcuts**

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 programmes. So try this:

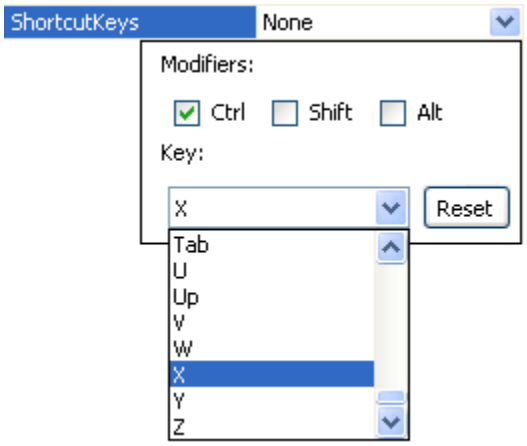
- In Design time, select the Exit item on your menu
- Look at the properties box on the right
- Locate the **ShortcutKeys** item:



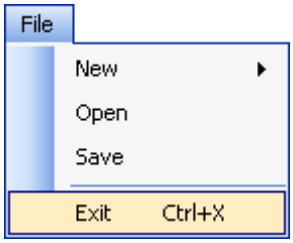

- Click the down arrow to reveal the following:



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:



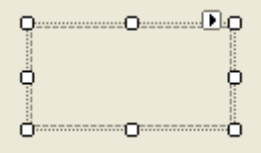

Click back on your menu to see what it looks like:




Run your programme 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 programme will close down.

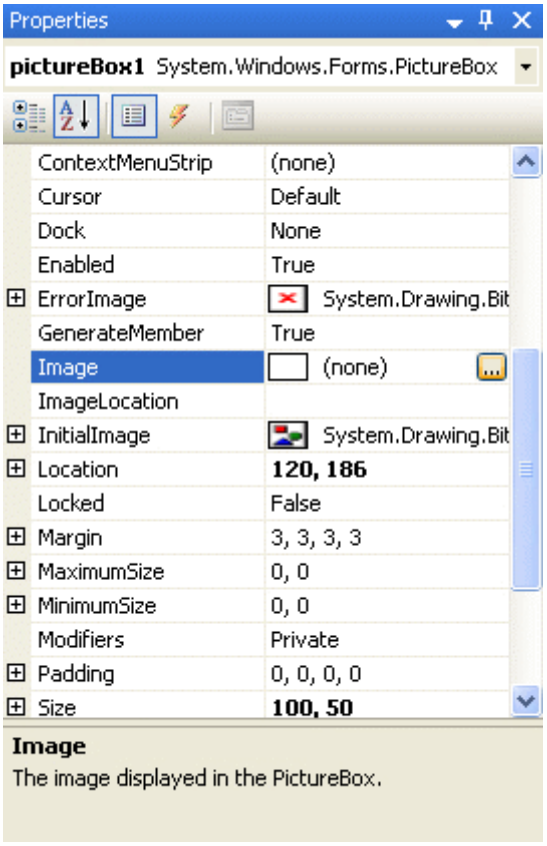
## The View Images menu Item

It's easy to add an image to your form with VB.Net. To insert an image, locate the **Picture** control in the toolbox. Either double click the control, or hold down your mouse on the form and draw one out. You should see something like this:

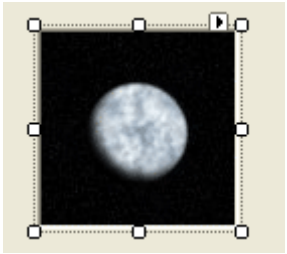


Change the **Height** and **Width** properties of the Picture Box to 100, 100. You'll have a small square. To make it stand out more, locate the **BorderStyle** property. Change the value to **Fixed3D**. Your Picture Box will then look like this:

To add a picture at design time, locate the **Image** property in the properties box:



Download and unzip the image at the top of the page. Then click the button with the three dots on it. A dialogue box appears. Locate an image. Select it, and then click Open in the dialogue box. The image will appear in your Picture Box:



If you select an image that is too big for the picture box, only part if it will be visible. The Picture Box control does not resize your image.

You can, however, set another property of the picture box - the **SizeMode** property. Set this to **AutoSize** and your picture box will resize to the size of your image.

## Check Boxes in VB .NET

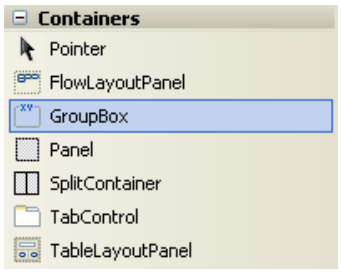
So start a new project. Locate the Checkbox control in the toolbox. Double click the control and a Checkbox appears on your new Form

You'll see that the Checkbox has the Text property of CheckBox1 by default, and a Name of CheckBox1. If you were to double click again on the Checkbox icon in the toolbox, the new control would be called CheckBox2.

The problem with this approach is that by double clicking each Checkbox, you have several individual Checkboxes. And if you wanted to move them around you'd have to move each Checkbox separately. There is a way to group all your Check Boxes together, and move them around as one - by using a **Group Box**.

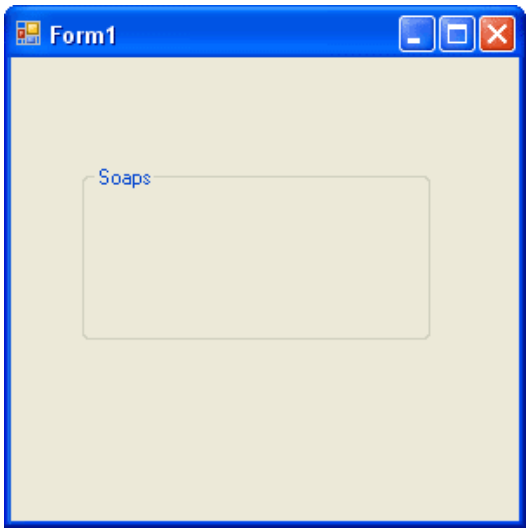
Click on your Checkbox with the right mouse button. From the menu that pops up, select delete to get rid of it.

Now locate the **Group Box** control in the toolbox:




It's better to draw this one on the form, rather than dragging and dropping. When you've added one, the only thing you should have on your Form is a Group Box.

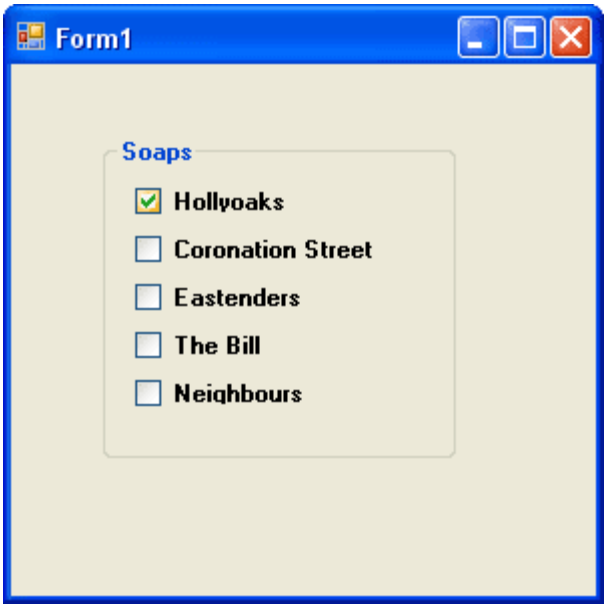
We're not going to be using many of the Properties in the Group Box Property box. But click on your Group Box to select it, and change to the Text Property to "Soaps". Change the Font Property to anything you like. You should now have a Form like this one



The Group Box we just added will hold our Checkboxes. It acts as a container for the controls. To move the Checkboxes about, we can just click on the Group Box to select it, and drag and drop the Group Box somewhere else. The Checkboxes will all move with the Group Box. Let's add some Checkboxes to it.

The only way to add a control to a Group Box is to draw one on the Group Box.

1. Click once with your left mouse button on the **Checkbox** icon in the VB toolbox
2. Move your mouse pointer over to the inside of the **Group Box**. The mouse pointer will change to a cross
3. Hold down you left mouse button inside the Group Box. Keep the button held down, and drag outwards. Release the left button when you're happy with the size. You can always resize it later.
4. Add 5 Checkboxes to your Group Box
5. Change the **Text** property of each of your Checkboxes to any five Soap Operas. Your Form should now look something like the one below:




Run your programme to test it out. Click inside a Checkbox to select an item. Click again to deselect it. When you've finished, return to the Design Environment and click on the Group Box itself to select it. Make sure the Group Box IS selected, and not one of your Checkboxes. You can now drag the Group Box around your Form and all the Checkboxes will move with it.

The point about having Checkboxes is to offer your users multiple choices.

## Add Option Buttons to a VB .NET form

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.

Adding Radio Buttons to a Form is exactly the same process as adding a Checkbox. Again, we'll add some Radio Buttons to a Group Box, and write code to extract what the user has chosen.

1. Add a **Group Box** to your Form.
2. Set the **Text** Property of the Group Box to "Best Sit Com of all time"
3. Set the **Font** options to anything you like
4. Place five **Radio Buttons** into your Group Box (By default, they'll be called "Option1", "Option2", "Option3", etc
5. Set the Text Property of the Five Radio Buttons to Only Fools and Horses, Dad's Army, Blackadder, Fawlty Towers, Vicar of Dibley
6. Your Form should now look something like this:


Run your programme and test to see if you can indeed only select one item from the list.

The reason you can only select one is that all the radio buttons are placed in the same group box. You can place another set of radio buttons in a second group box, and these would work independently of the set of radio buttons in the first group box.

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 5 - User Interface Development using Microsoft Publisher

By Mohd Taib bin Wahid

Fall 2011-2012

# Lab 5: User Interface Development using Microsoft Publisher

Objectives

- Understanding of the user interface development using Microsoft Publisher

Activity Outcomes

- The students will know how to develop web site interface by using Microsoft Publisher

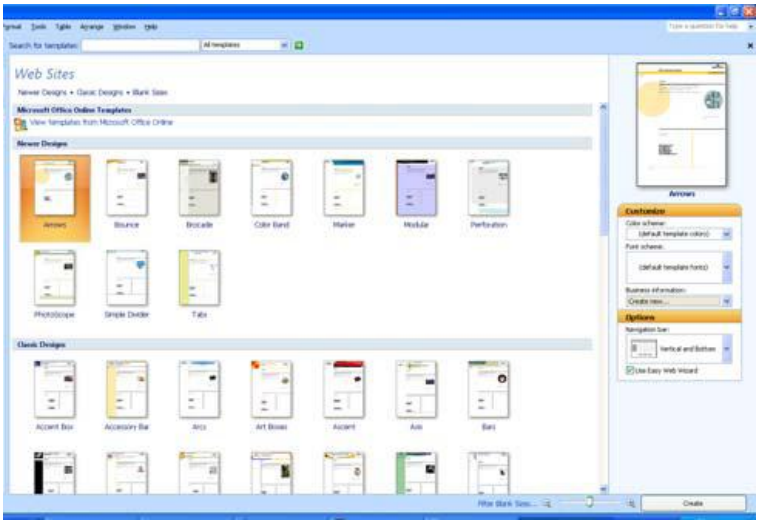
- Understanding of the user interface development using Mivrosoft Publisher

## Creating a Web Site with Microsoft Publisher 2007

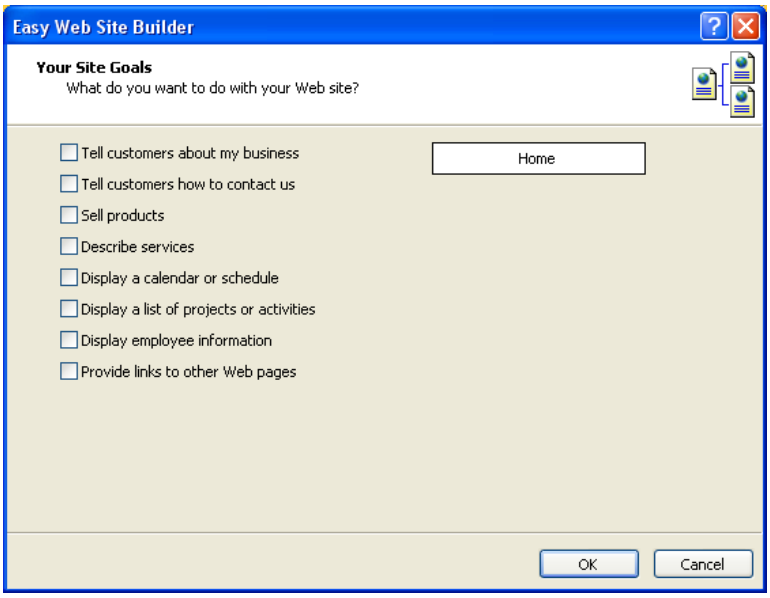
If you want a way to create a simple web site with text and pictures, Microsoft Publisher is your answer. Publisher allows you to create web pages without having to know HTML coding. When you are ready to publish your site, Publisher assembles all of the related files into a single folder. You can then upload your web site to NetDrive which will be used as your hosting service.

**To Create a New Web Site**
1) Open **Publisher**
2) Click **Web Sites**


3) Click on a **design template**
4) Click **Create**




5) In the **Easy Web Site Builder** select the type of web pages you want added to your web site
6) Click **OK**




A navigation bar is included on all of the pages within the web site and automatically updates when new pages are added. The first page to appear is the **Home** page. This is the first page that will be seen when your web site is viewed.

**To Change the Title or the Logo on all Pages**
1) Click on **Edit**
2) Click **Business Information**
3) Type the Organization Name, which is used as the title on all of the web pages
4) Change the logo if you wish to include one on all the pages

**Modifying Web Pages**
Your publication is made up of text frames and picture frames. You replace the text and pictures with your own text and pictures, or with other objects.

**To Replace Text**:
1) Click within the text frame
2) Type your text

**To Replace Picture Frames:**
1) Click on the picture to select it
2) Click on **Insert**
3) Click **Picture**
4) Click on **Clip Art** or **From File**

**To Insert a New Text Box:**
1) On the Objects toolbar, click **Text Box**
2) In your publication, click and drag to draw a new text frame
3) Type your text

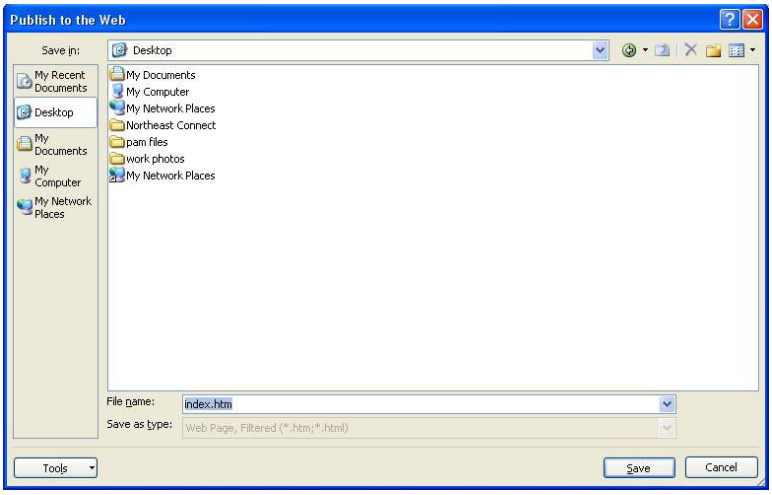
**To Insert a Picture:**
1) Click on **Insert**
2) Click **Picture**
3) Click on **Clip Art** or **From File**
4) Once the picture is inserted you can resize it and/or move it

**To Save the Publication**
1) Click on **File**
2) Click on **Save**

This is being saved as a publication file. This is the file you will open when changes need to be made to the web site.

**To Publish your Web Site**
1) Click on **View**
2) Click on **Toolbars**
3) Click on **Web Tools**
4) Click on **Publish to Web**


5) Click **OK** for the **Publish to Web** message
6) Choose a location to save and leave the file name as (index.htm). Index.htm determines that this is the first page of the web site.
7) Click **Save**




8) Click **OK** for the below message:

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 6 Prototypes

By Mohd Taib bin Wahid

Fall 2011-2012

Objectives

- Understanding of the importance and working on prototypes in project
- Understand why we use prototype for explaining your project more and just go through all the events and windows we can have at the time of implantation phase

Activity Outcomes

- The students will know the concept of prototype and how to generate them
- They will know what are the types of prototypes we can have and how we can deal with different types of prototypes

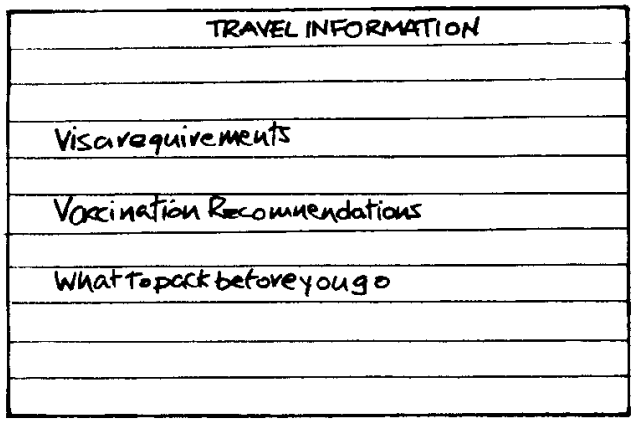
- Understanding of the importance and working on prototypes in project
- Understand why we use prototype for explaining your project more and just go through all the events and windows we can have at the time of implantation phase

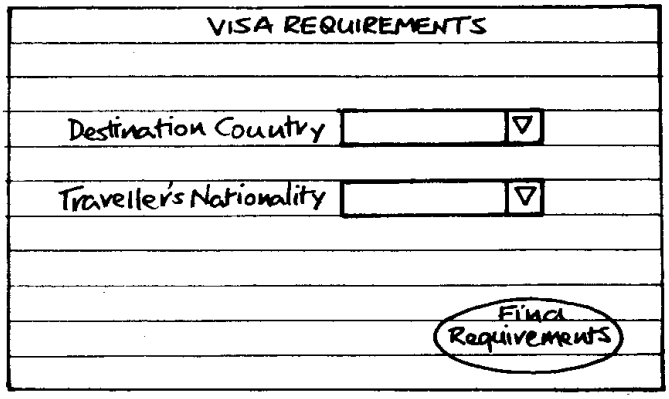
What is Prototype?

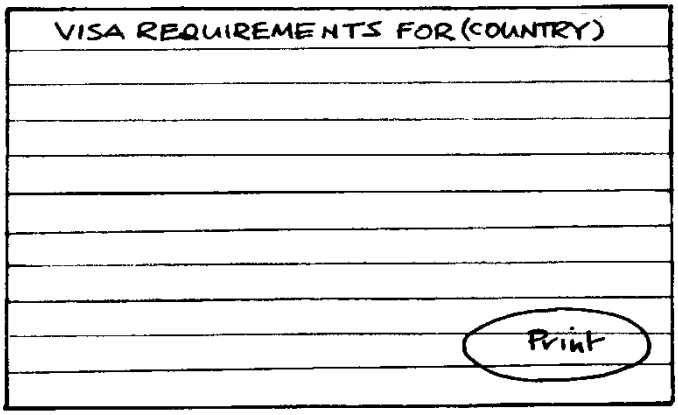
Why Prototype use?

Different kind of prototyping discussing in the lab.

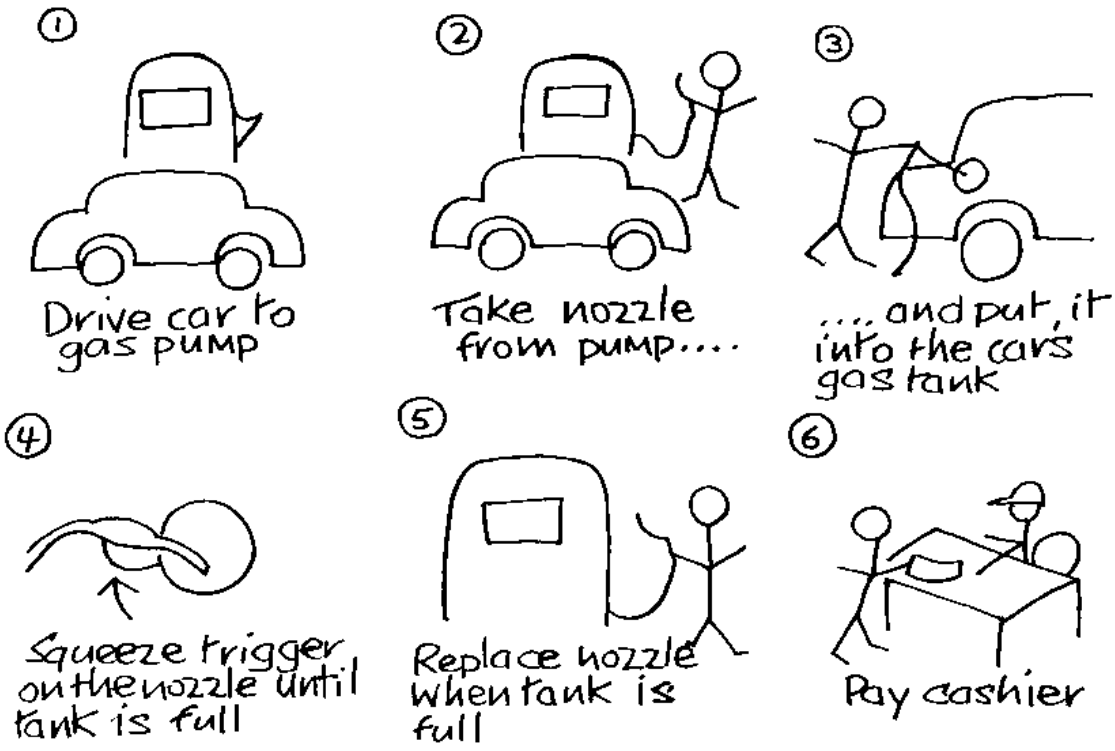
A series of screen sketches



A storyboard (A Cartoon Like Series of Scenes)



Practice Questions:

Follow the list of different scenario, draw Series of Sketches and A story Board prototypes for each scenario on the paper by using pen or pencil:

1. Steps of using ATM Machine for withdraw or deposit money.(Apply Series of Sketches)
2. Steps of Purchasing some items from a shopping center. (A story Board)
3. Steps of Installing any software on computer machine. (Both)

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 7 Design Rules

By Mohd Taib bin Wahid

Fall 2011-2012

# Lab 7: Design Rules

Objectives

- Understanding of the beneficial of designing rules (principles, standards, and guidelines) on the interface designing

Activity Outcomes

- The students will know what is the difference between principles, standards, and guidelines
- They will learn how to apply the principles, standards, and guidelines into the graphical user interface

**Q1.** Distinguish between principles, guidelines and standards, using examples of each to illustrate?

**Answer 1:**

**Q2.** Can principles be used to prove the usability of a system? Justify your answer.

**Answer 2:**

**Q3.** On the basis of your own experience of using a typical bank cash machine (ATM), and after learning the Jakob Nielsen's 10 Heuristics Principles, describe what designer should and should not do.

**Jakob Nielsen's 10 Heuristics Principles**

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

**Answer 3:**

1. **Visibility of system status:** ATM should give an immediate feedback once users entered valid or invalid password. (If password is valid, ATM provide welcome message, if password is invalid display error message with solution).

2. **Match between system and the real world**

3. **User control and freedom**

4. **Consistency and standards**

5. **Error prevention**

6. Recognition rather than recall

7. Flexibility and efficiency of use

8. Aesthetic and minimalist design

9. Help users recognize, diagnose, and recover from errors

10. Help and documentation

KING ABDULAZIZ UNIVERSITY
Faculty of Computing and Information Technology

# CPIS - 354

## Lab Manual: Lab 8 - Evaluation Method

By Mohd Taib bin Wahid

Fall 2011-2012

# Lab 8: Evaluation Method

Objectives

- Understanding of the Evaluation Method on the interface designing

Activity Outcomes

- The students will know that what are the evaluation methods and how to apply

# Lab 8: Evaluation Method

## Exercise 1:

Choose an appropriate **Evaluation Method** for each of the following situations. In each case suggest the answers in the given spaces below:

**(a) You are at an early stage in the design of a spreadsheet package and you wish to test what type of icons will be easiest to learn.**

(i) Participants _____

(ii) The technique used_____

(iii) Representative tasks to be examined_____

(iv) Measurements that would be appropriate_____

(v) An outline plan for carrying out the evaluation_____

**(b) You have designed and implemented a new game system and want to evaluate it before release**

(i) Participants _____

(ii) The technique used_____

(iii) Representative tasks to be examined_____

(iv) Measurements that would be appropriate_____

(v) An outline plan for carrying out the evaluation_____

(c) You have been asked to develop a system to store and manage student exam results and would like to test two different designs prior to implementation or prototyping.

(i) Participants _____

(ii) The technique used_____

(iii) Representative tasks to be examined_____

(iv) Measurements that would be appropriate_____

(v) An outline plan for carrying out the evaluation_____