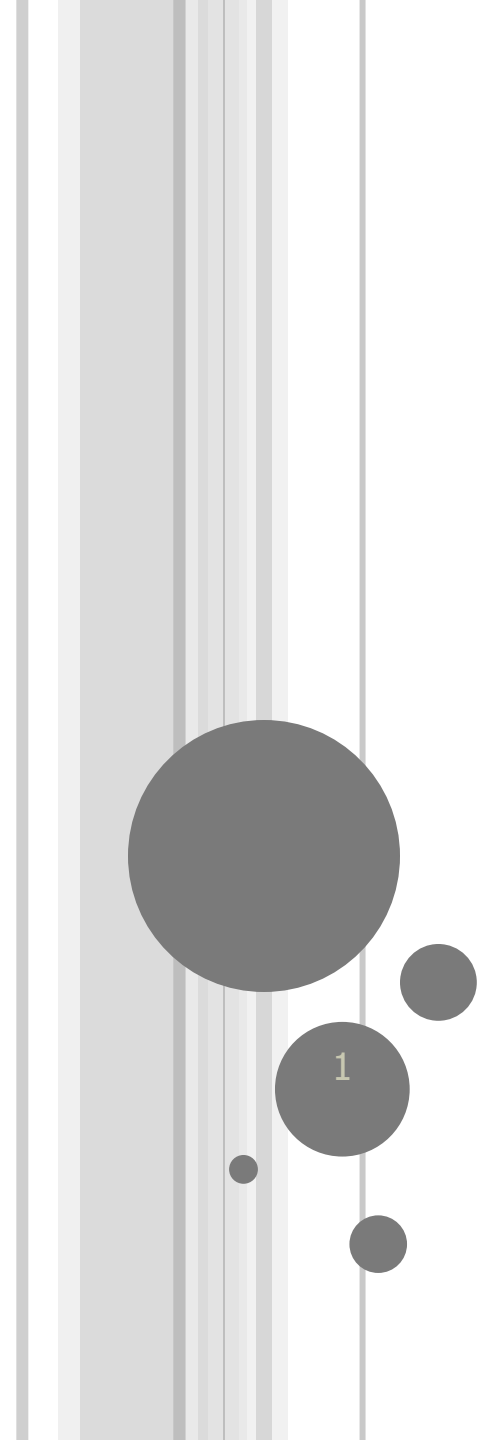




REVIEW OF CHAPTER 1

1

- Trouble installing/accessing Visual Studio?

- Computer
 - a device that can perform calculations and make logical decisions much faster than humans can
- Computer programs
 - a sequence of instructions
 - Input—Process—Output
(data—manipulate—results)
- Hardware: physical components (keyboard, screen, mouse, hard drive, memory, DVDs, CPU—Central Processing Unit)
- Software: programs that run on a computer

- Programmers vs. Users
- Computer languages:
 - Machine language: computers understand.
 - High-level language: resembles everyday English. Compilers convert it into machine language. (Visual Basic, C, C++, Java, C#, Fortran, COBOL, ...)

- Algorithm: a step by step series of instructions to solve a problem
- Visual Basic.NET is:
 - Event-driven
 - Object-oriented
 - User-friendly (GUI-based)

CHAPTER 2

Visual Basic, Controls, and Events

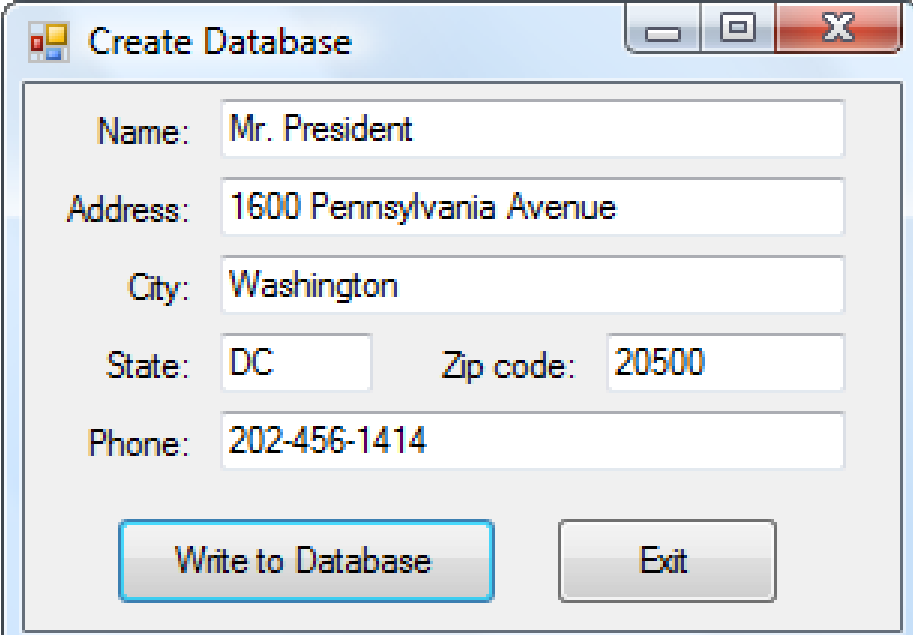
VISUAL BASIC, CONTROLS, AND EVENTS

- 2.1 An Introduction to Visual Basic
- 2.2 Visual Basic Controls
- 2.3 Visual Basic Events

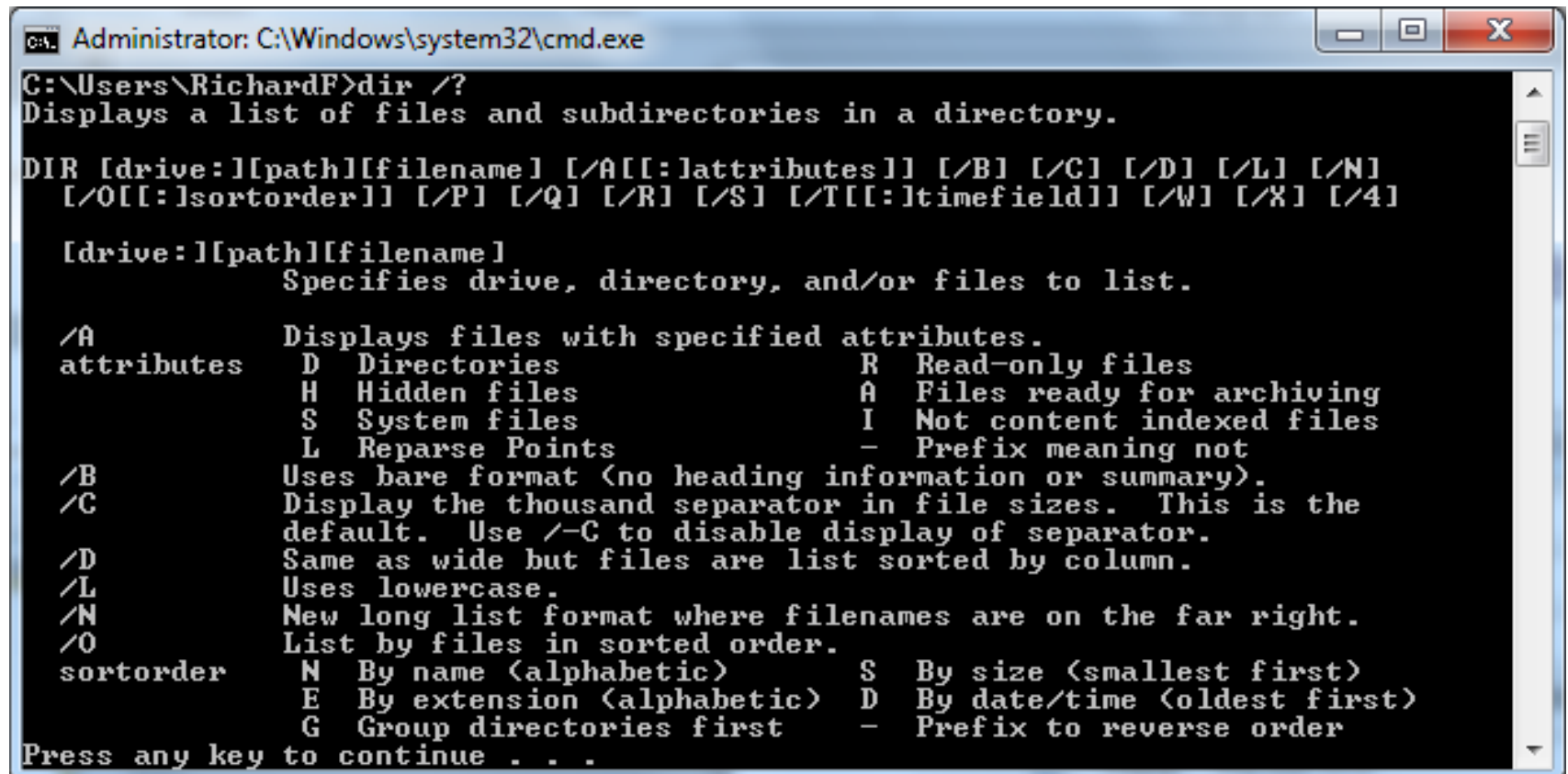
- Before VB, a window required hundreds of lines of code
- Now
 - Fewer instructions
 - Consistent
 - Less time
 - User-friendly

VISUAL BASIC 2010

- Language used to create Windows applications.
- Provides a Graphical User Interface or GUI.
- The sequence of instructions executed in the program is controlled by events.



The screenshot shows a Windows-style application window titled "Create Database". The window has a standard title bar with minimize, maximize, and close buttons. The main content area contains several text input fields with labels to their left: "Name:" with the value "Mr. President", "Address:" with "1600 Pennsylvania Avenue", "City:" with "Washington", "State:" with "DC", "Zip code:" with "20500", and "Phone:" with "202-456-1414". At the bottom of the window, there are two buttons: "Write to Database" and "Exit".



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\RichardF>dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N]
  [/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

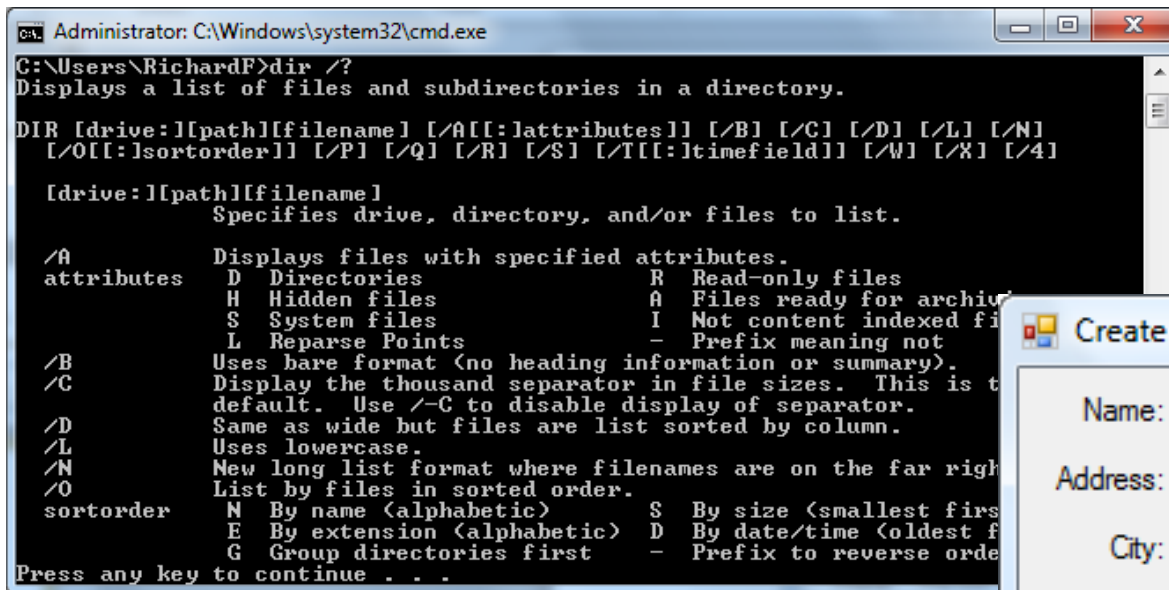
[drive:][path][filename]
    Specifies drive, directory, and/or files to list.

/A      Displays files with specified attributes.
attributes  D Directories                R Read-only files
              H Hidden files              A Files ready for archiving
              S System files              I Not content indexed files
              L Reparse Points            - Prefix meaning not

/B      Uses bare format (no heading information or summary).
/C      Display the thousand separator in file sizes. This is the
        default. Use /-C to disable display of separator.
/D      Same as wide but files are list sorted by column.
/L      Uses lowercase.
/N      New long list format where filenames are on the far right.
/O      List by files in sorted order.
sortorder  N By name (alphabetic)         S By size (smallest first)
              E By extension (alphabetic) D By date/time (oldest first)
              G Group directories first   - Prefix to reverse order

Press any key to continue . . .
```

- Program complicated, unfriendly
- Required parameters
- Once input was entered, hard to go back



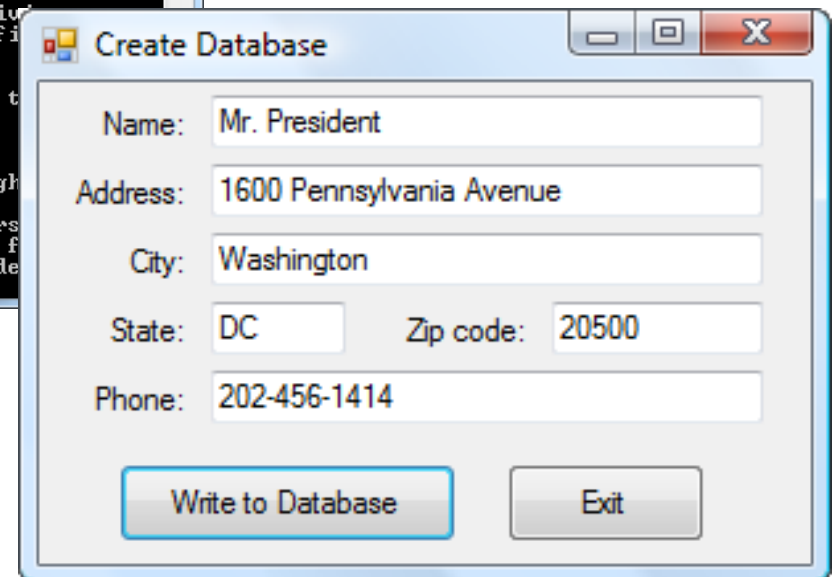
```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\RichardF>dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N]
[/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

[drive:][path][filename]
    Specifies drive, directory, and/or files to list.

/A      Displays files with specified attributes.
attributes  D Directories          R Read-only files
              H Hidden files        A Files ready for archiving
              S System files         I Not content indexed files
              L Reparse Points       - Prefix meaning not
/B      Uses bare format (no heading information or summary).
/C      Display the thousand separator in file sizes. This is the
default. Use /-C to disable display of separator.
/D      Same as wide but files are list sorted by column.
/L      Uses lowercase.
/N      New long list format where filenames are on the far right.
/O      List by files in sorted order.
sortorder  N By name (alphabetic)    S By size (smallest first)
              E By extension (alphabetic) D By date/time (oldest first)
              G Group directories first - Prefix to reverse order

Press any key to continue . . .
```



Create Database

Name: Mr. President

Address: 1600 Pennsylvania Avenue

City: Washington

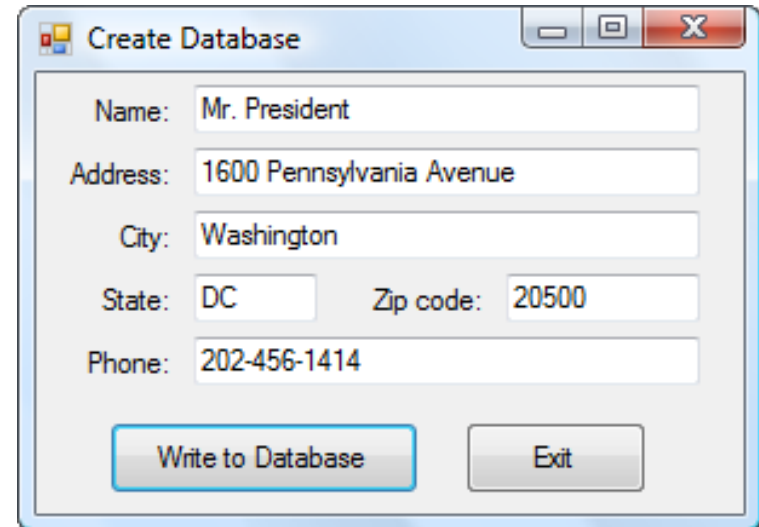
State: DC Zip code: 20500

Phone: 202-456-1414

Write to Database Exit

HOW TO DEVELOP A VB APPLICATION

- Design the Interface for the user
 - Literally draw the GUI
 - Drag buttons/text boxes/etc onto form
- Determine which events the controls on the window should recognize
- Write the code for those events.
- Test



The screenshot shows a Windows-style application window titled "Create Database". The window contains a form with the following fields and controls:

- Name:
- Address:
- City:
- State: Zip code:
- Phone:
- Buttons: "Write to Database" and "Exit"

WHAT HAPPENS WHEN PROGRAM IS RUNNING

1. VB monitors the controls for events
2. If event occurs, it runs procedures assigned to that event
3. If no event exists, it goes back to #1

DIFFERENT VERSIONS OF VISUAL BASIC

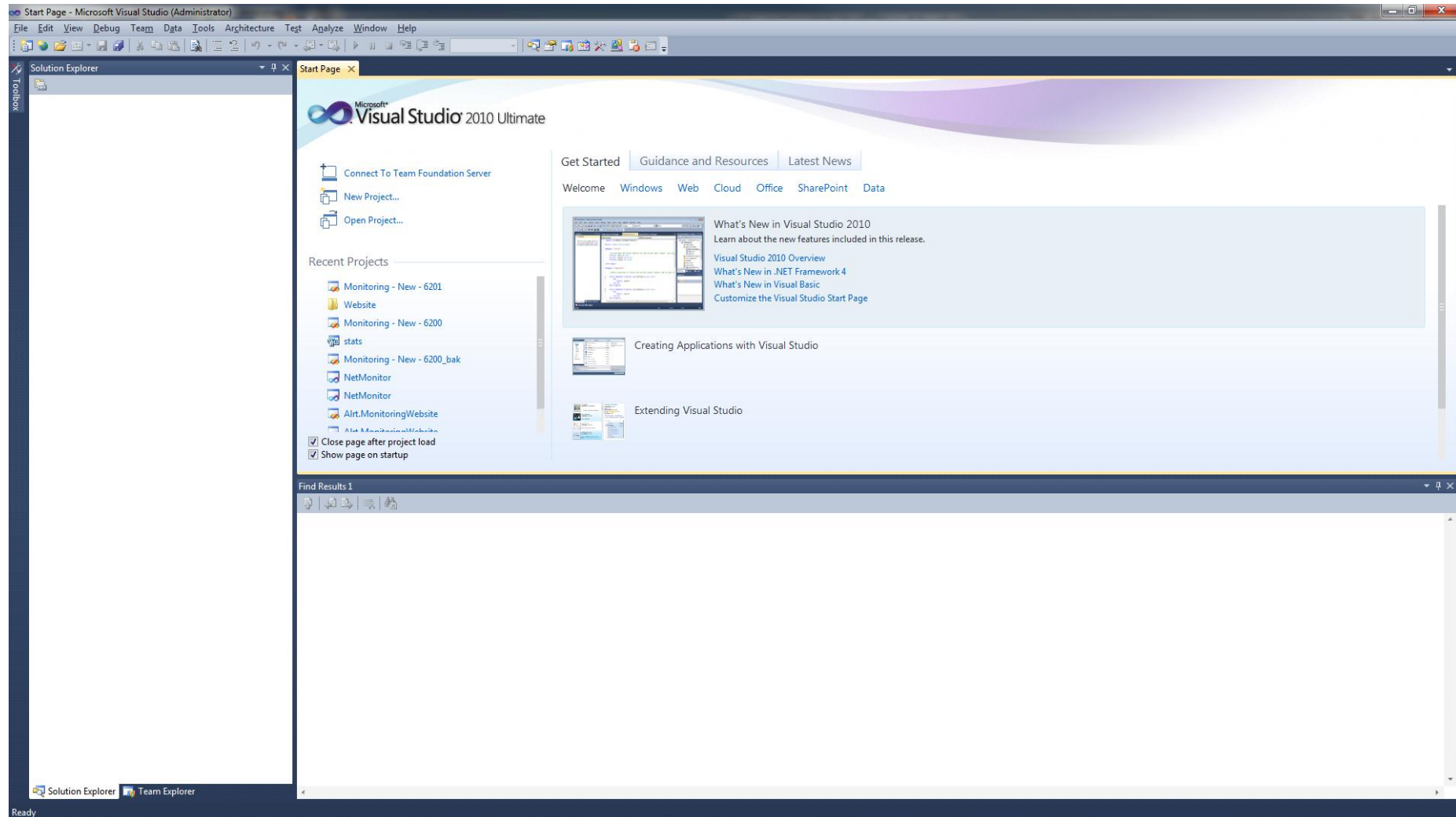
- Version 1.0 – 1991 Version 2.0 – 1992
- Version 3.0 – 1993 Version 4.0 – 1995
- Version 5.0 – 1997 Version 6.0 – 1998

- Visual Basic.NET – 2002 (NOT BACKWARD COMPATIBLE WITH EARLIER VERSIONS)
- Visual Basic 2005 – November 2005
- Visual Basic 2008 – November 2007
- Visual Basic 2010 – April 2010
- Visual Basic 2012 – Sept 2012

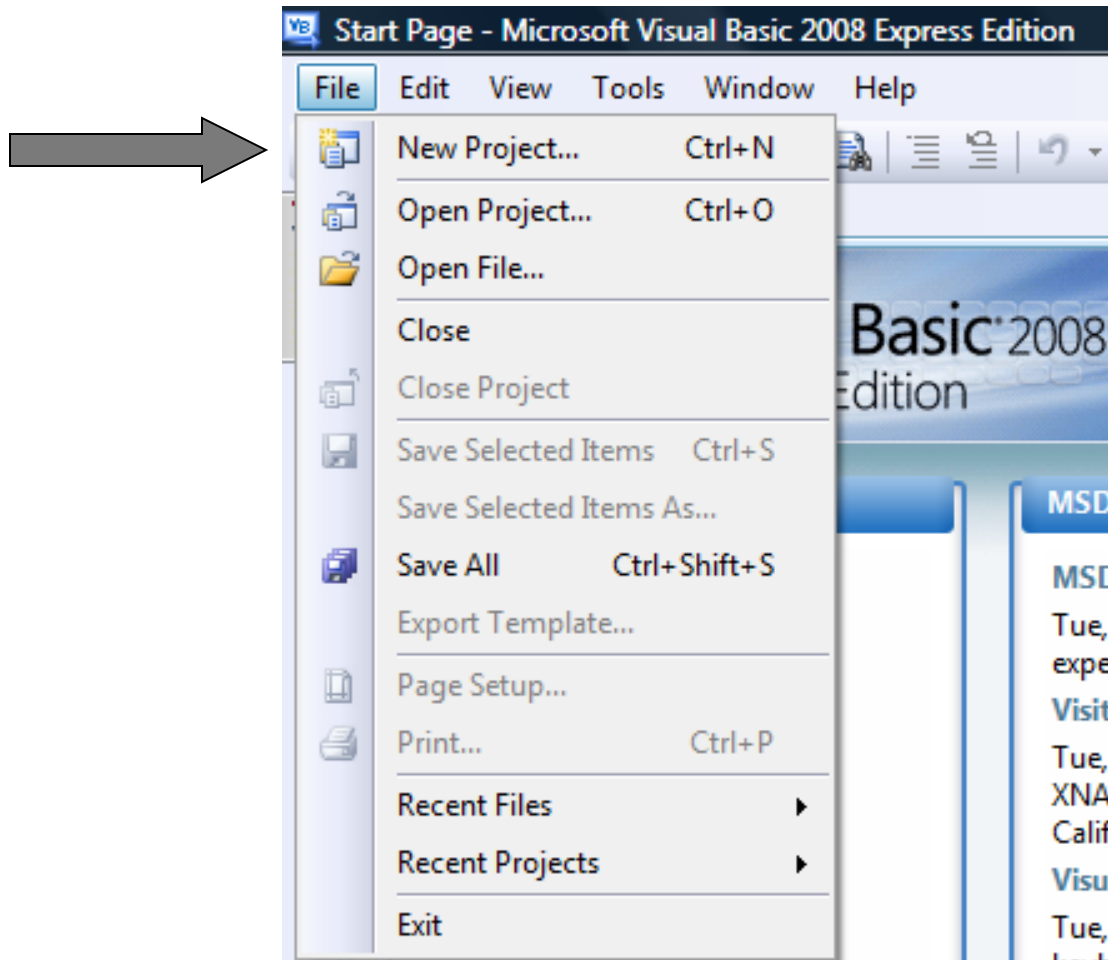
2.2 VISUAL BASIC CONTROLS

- Invoking Visual Basic
- Text Box Control
- Button Control
- Label Control
- List Box Control
- Name Property
- Help / Fonts / Auto Hide
- Positioning and Aligning Controls

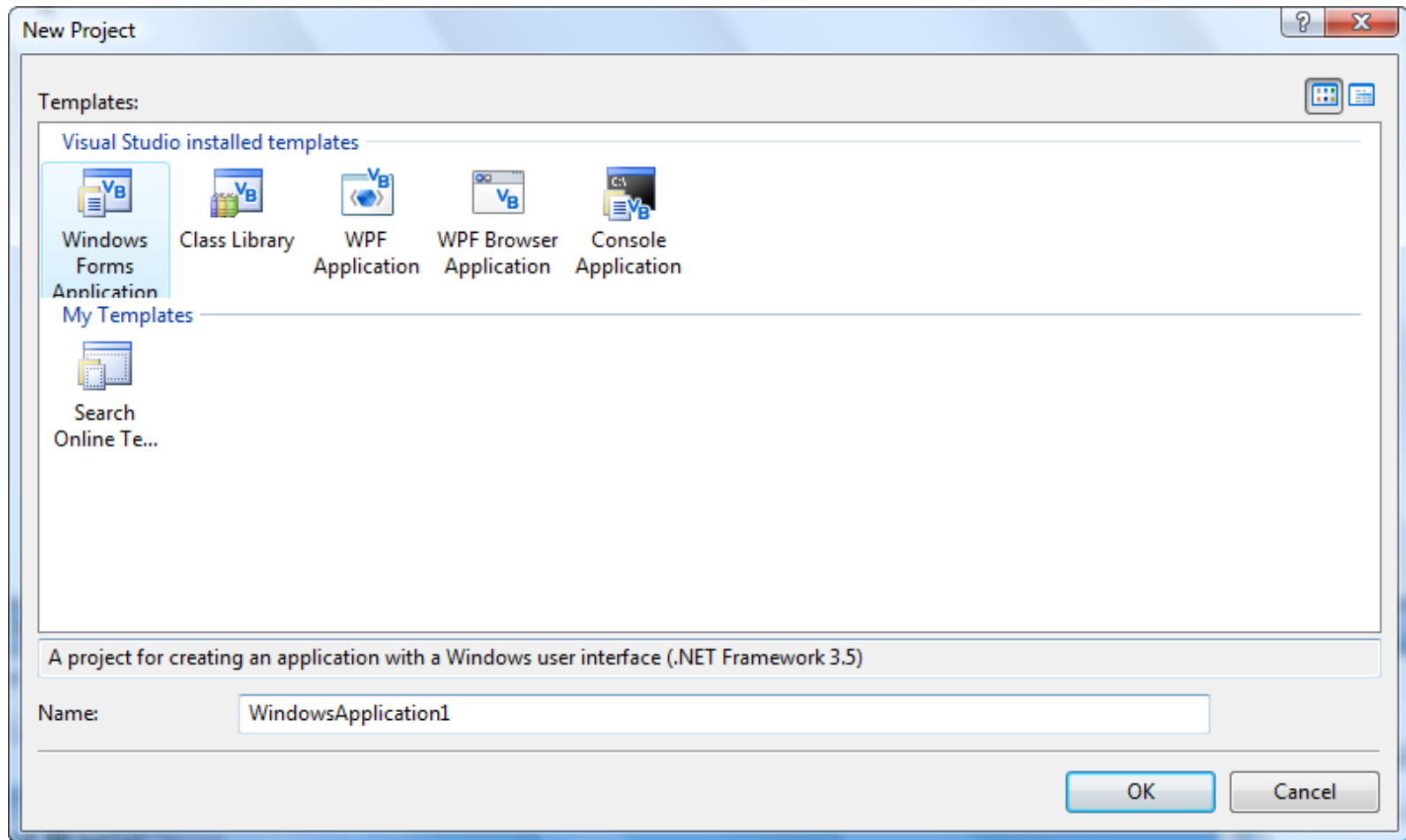
VISUAL BASIC START PAGE



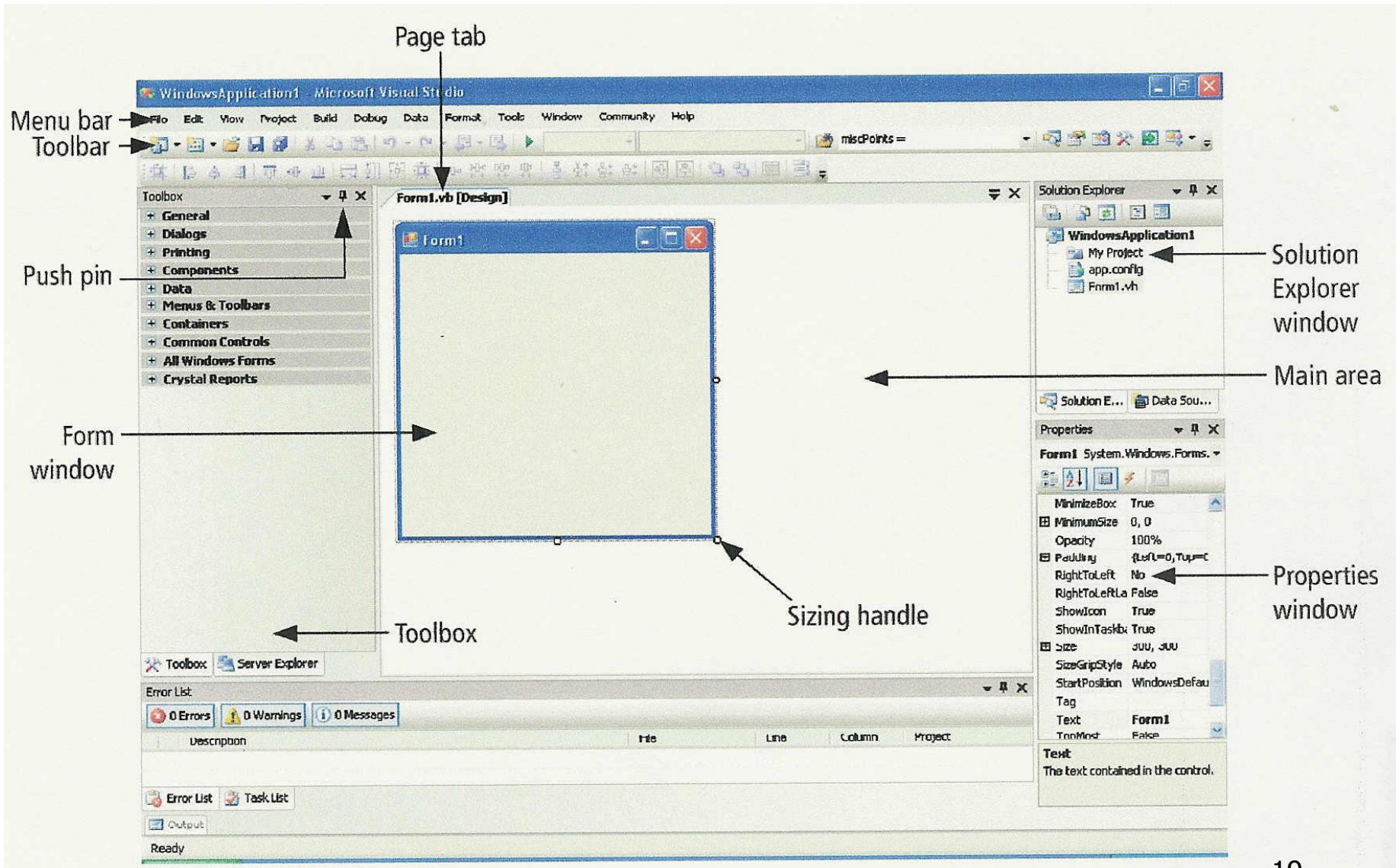
START A NEW PROJECT

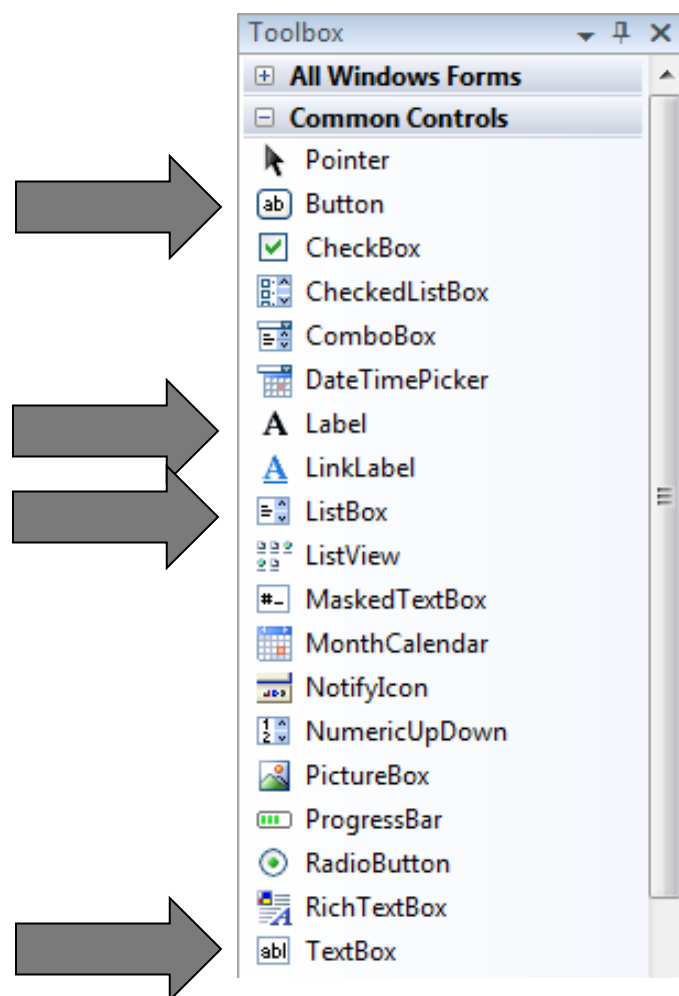


NEW PROJECT DIALOG BOX

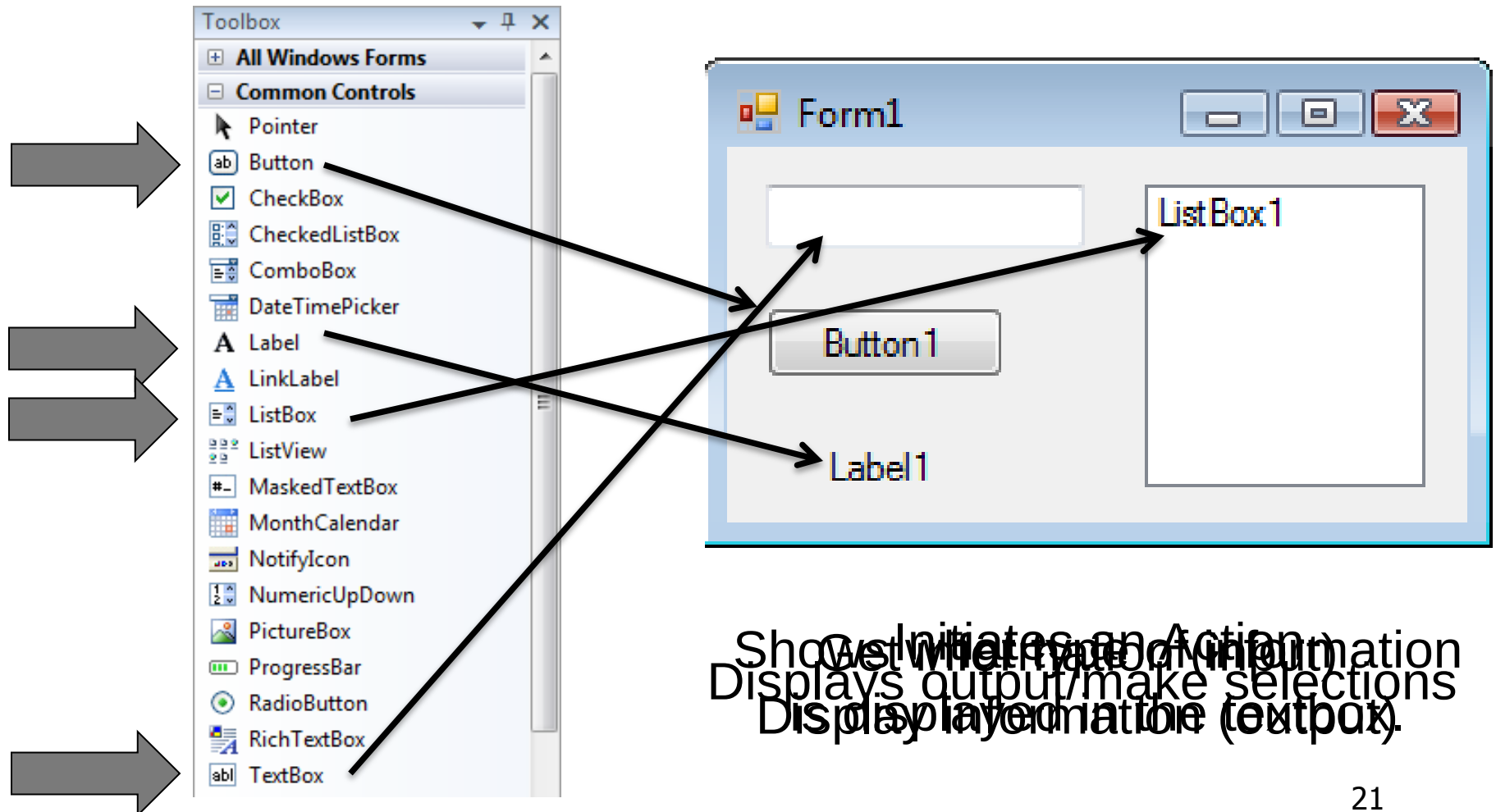


INITIAL VISUAL BASIC SCREEN





TOOLBOX

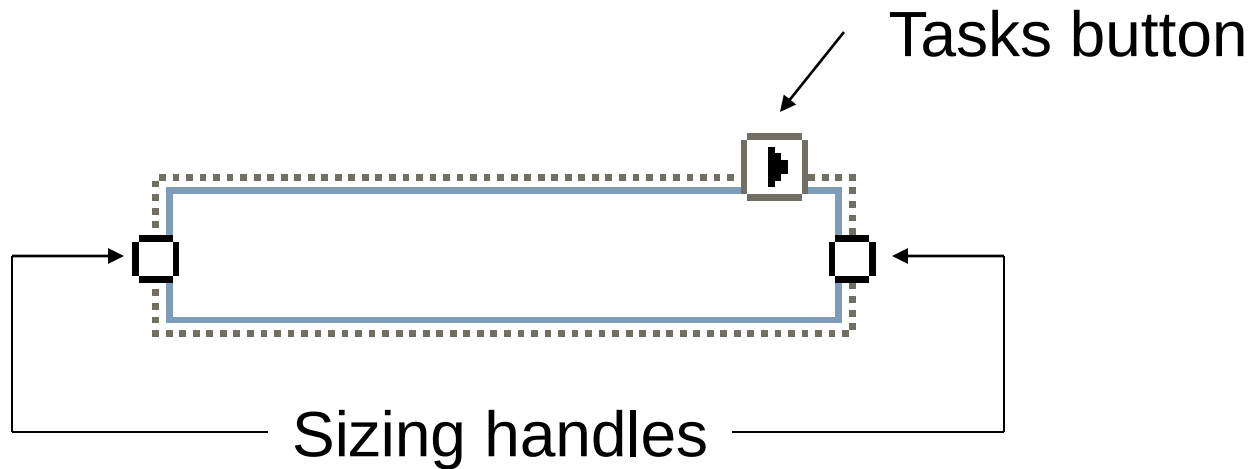


PLACING CONTROL FORMS

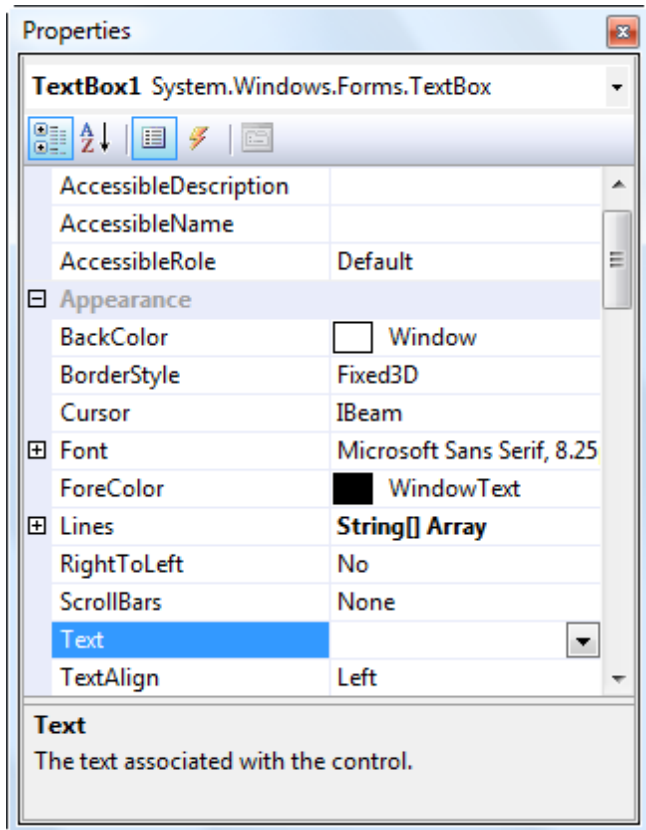
- 3 Ways to Place a Control from the Toolbox onto the Form Window
 - Double-click
 - Drag and Drop
 - Click, Point, and Drag

TEXT BOX CONTROL

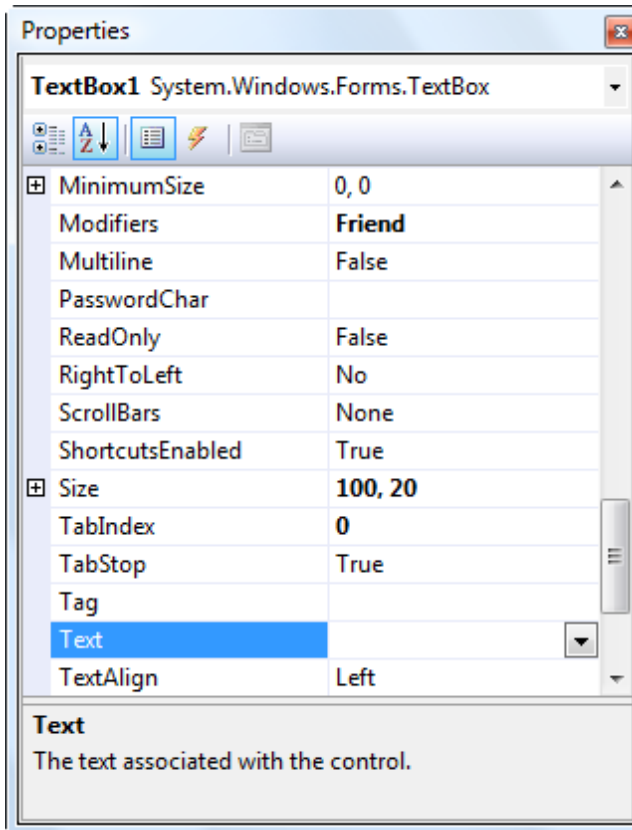
- Used for input and output
- When used for output, ReadOnly property is set to True



PROPERTIES WINDOW



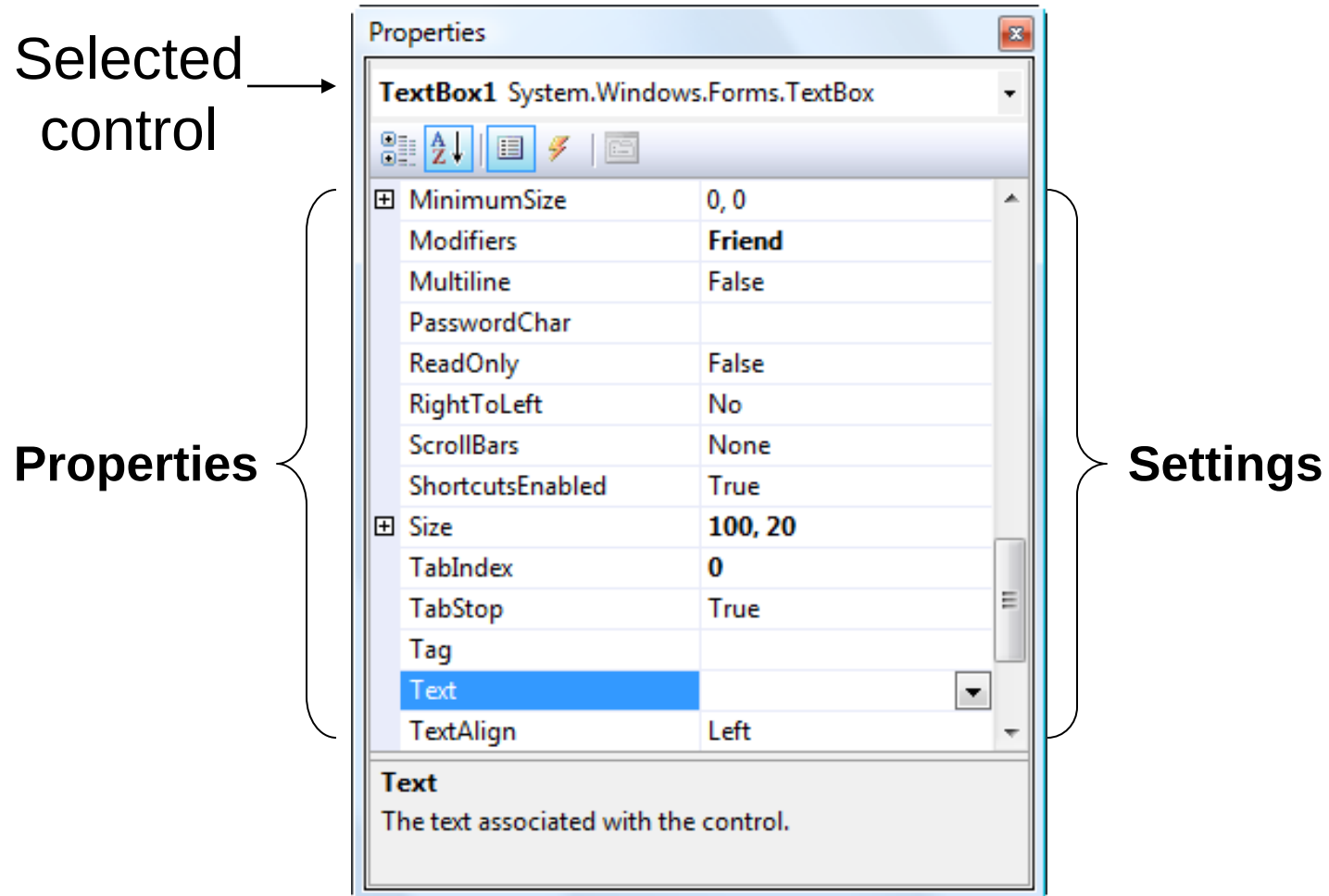
Categorized view

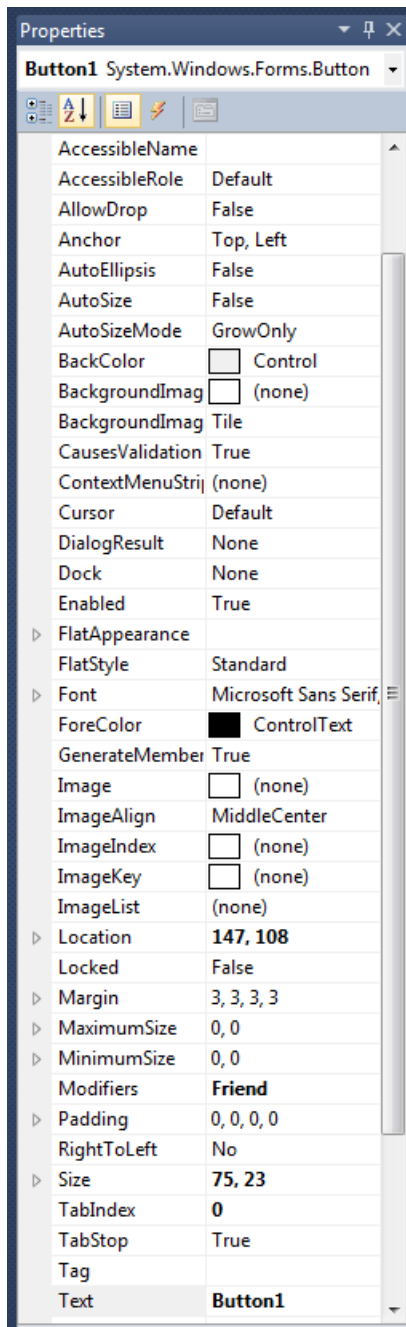


Alphabetical view

Press F4 to display the Properties window for the selected control.

PROPERTIES WINDOW



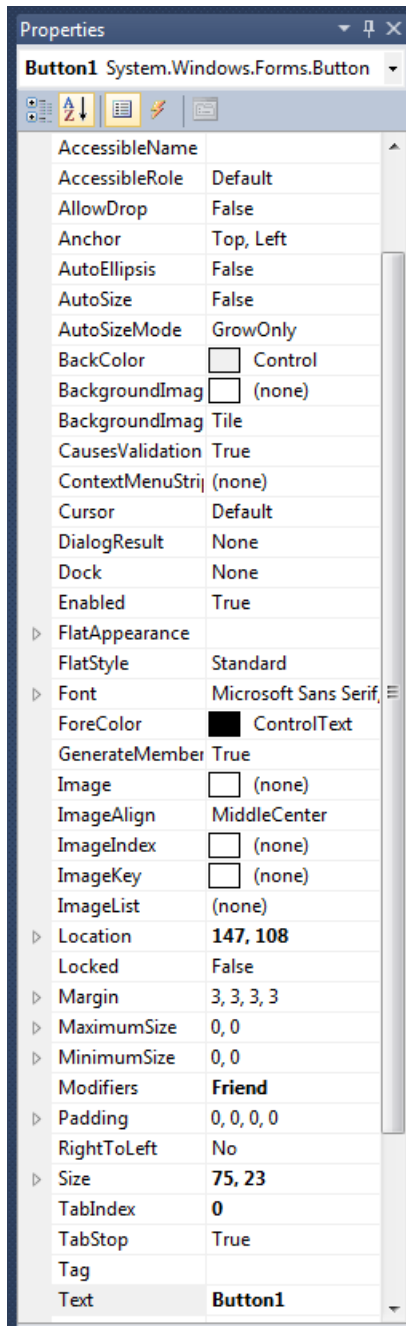


SOME OFTEN USED PROPERTIES

- Text
- Autosize
- Font.Name
- Font.Size
- ForeColor
- BackColor
- ReadOnly

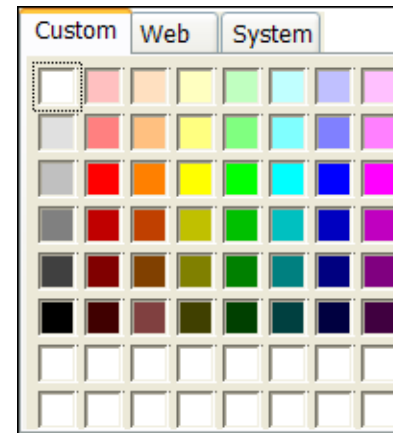
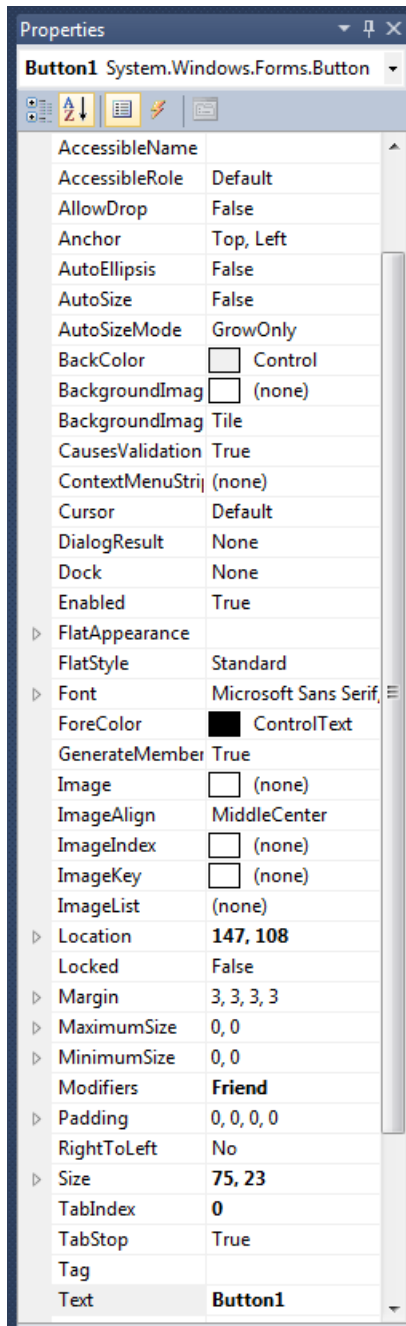
SETTING PROPERTIES

- Click on property name in left column.
- Enter its setting into right column by typing or selecting from options displayed via a button or ellipses.



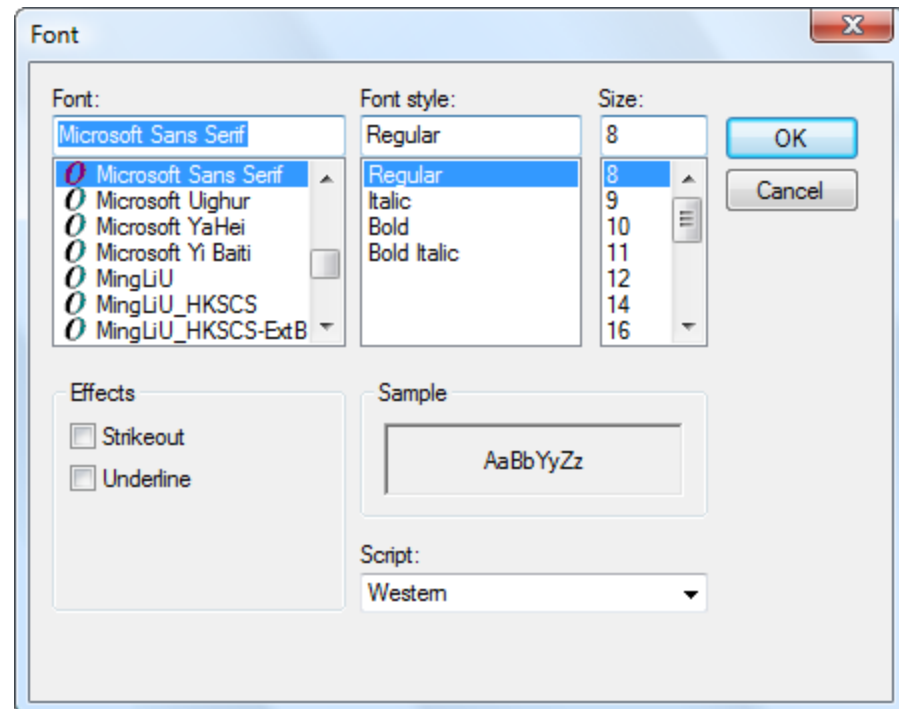
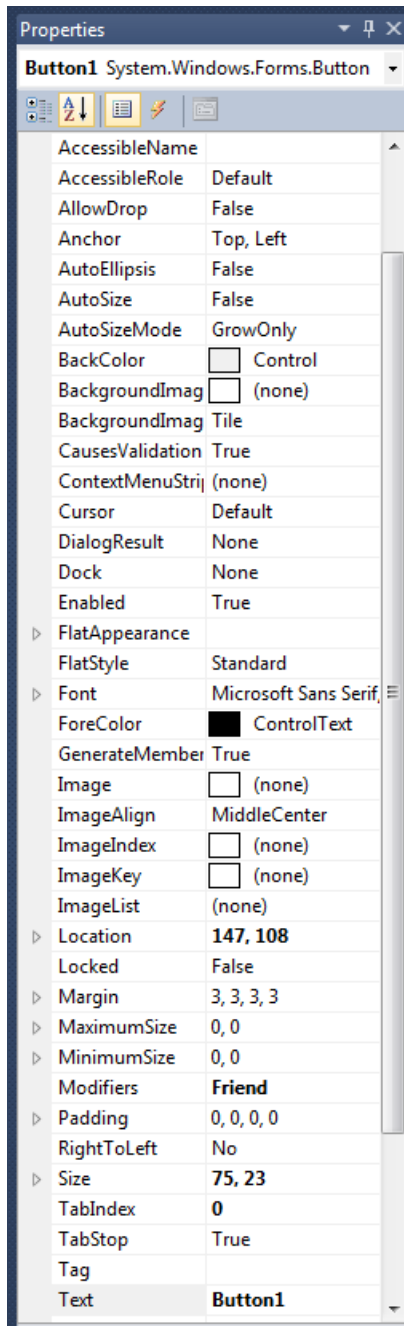
ForeColor PROPERTY

1. Click on ForeColor.
2. Click on button at right of settings box.
3. Click on Custom tab to obtain display shown.
4. Click on a color.



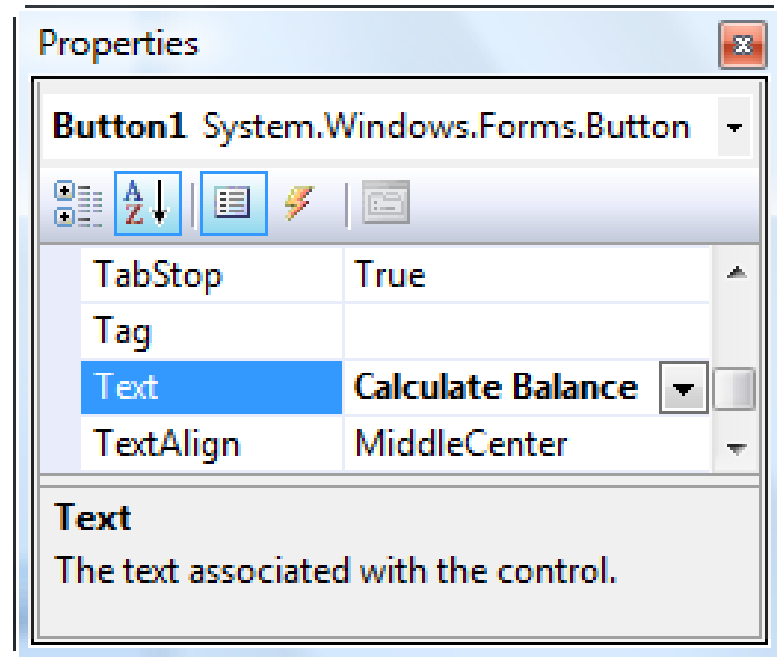
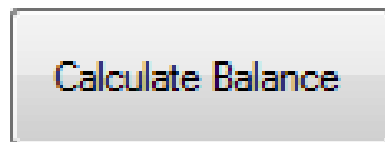
FONT PROPERTY

1. Click on Font in left column.
2. Click on ellipsis at right of settings box to obtain display shown,
3. Make selections.



BUTTON CONTROL

- The caption on the button should indicate the effect of clicking on the button.
- Text property determines caption.



ADD AN "ACCESS KEY"

Calculate Balance

Calculate Balance

Properties

Button1 System.Windows.Forms.Button

TabStop True

Tag

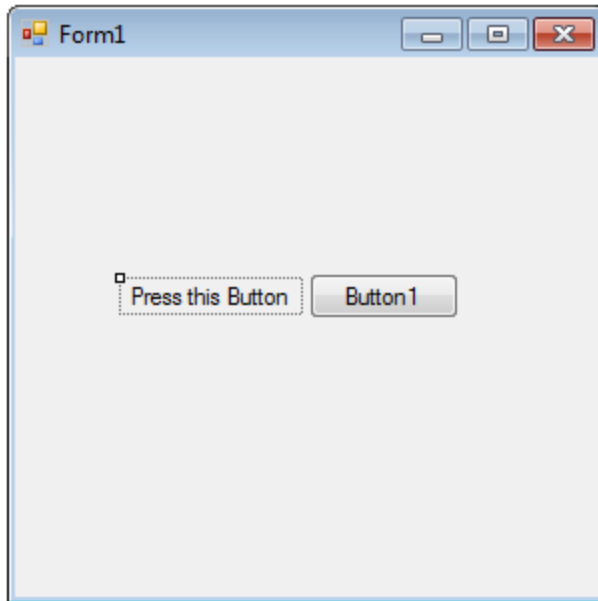
Text &Calculate Balance

TextAlign MiddleCenter

Text
The text associated with the control.

LABEL CONTROL

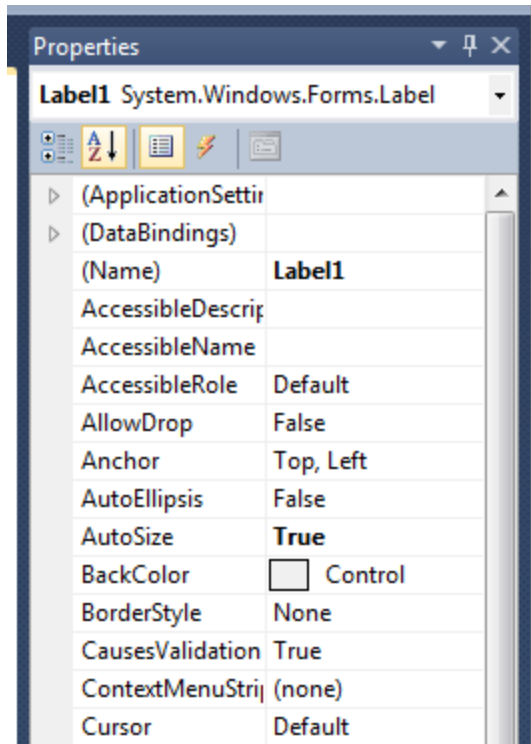
- Used to identify the contents of a text box/button/etc
- Text property specifies caption
- By default, label automatically resizes to accommodate caption on one line
- When the AutoSize property is set to False, label can be resized manually. Used primarily to obtain a multi-rowed label



LIST BOX CONTROL

- Initially used to display several pieces of output.
- In Chapter 9 used to select from a list.

THE NAME PROPERTY



- Used by the programmer to refer to a control in code
- Setting for Name property near top of Properties window.
- Name must begin with a letter, be less than 215 characters long, and may include numbers and letters.
- Use appropriate 3- or 4-character naming prefix

CONTROL NAME PREFIXES

Control	Prefix	Example
button	btn	btnCompute
label	lbl	lblAddress
text box	txt	txtAddress
list box	lst	lstOutput

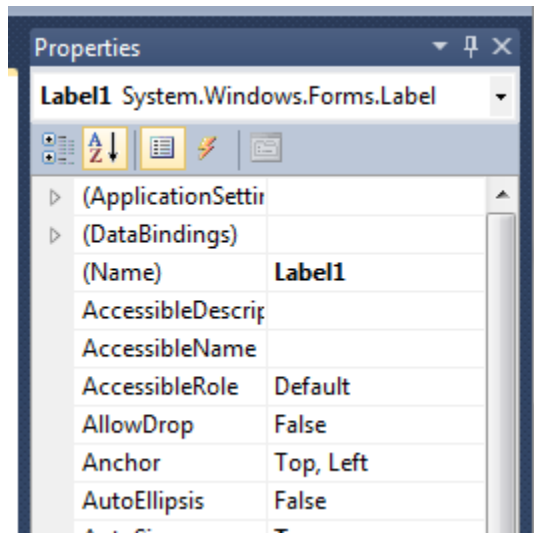
RENAMING THE FORM

- Initial name is Form1
- The Solution Explorer window lists a file named Form1.vb
- To rename the form, change the name of this file to *newName.vb*
- *newName* should begin with prefix *frm*

- Proportional width fonts take up less space for "I" than for "W" – like Microsoft Sans Serif
- Fixed-width fonts take up the same amount of space for each character – like Courier New
- Fixed-width fonts are used for tables

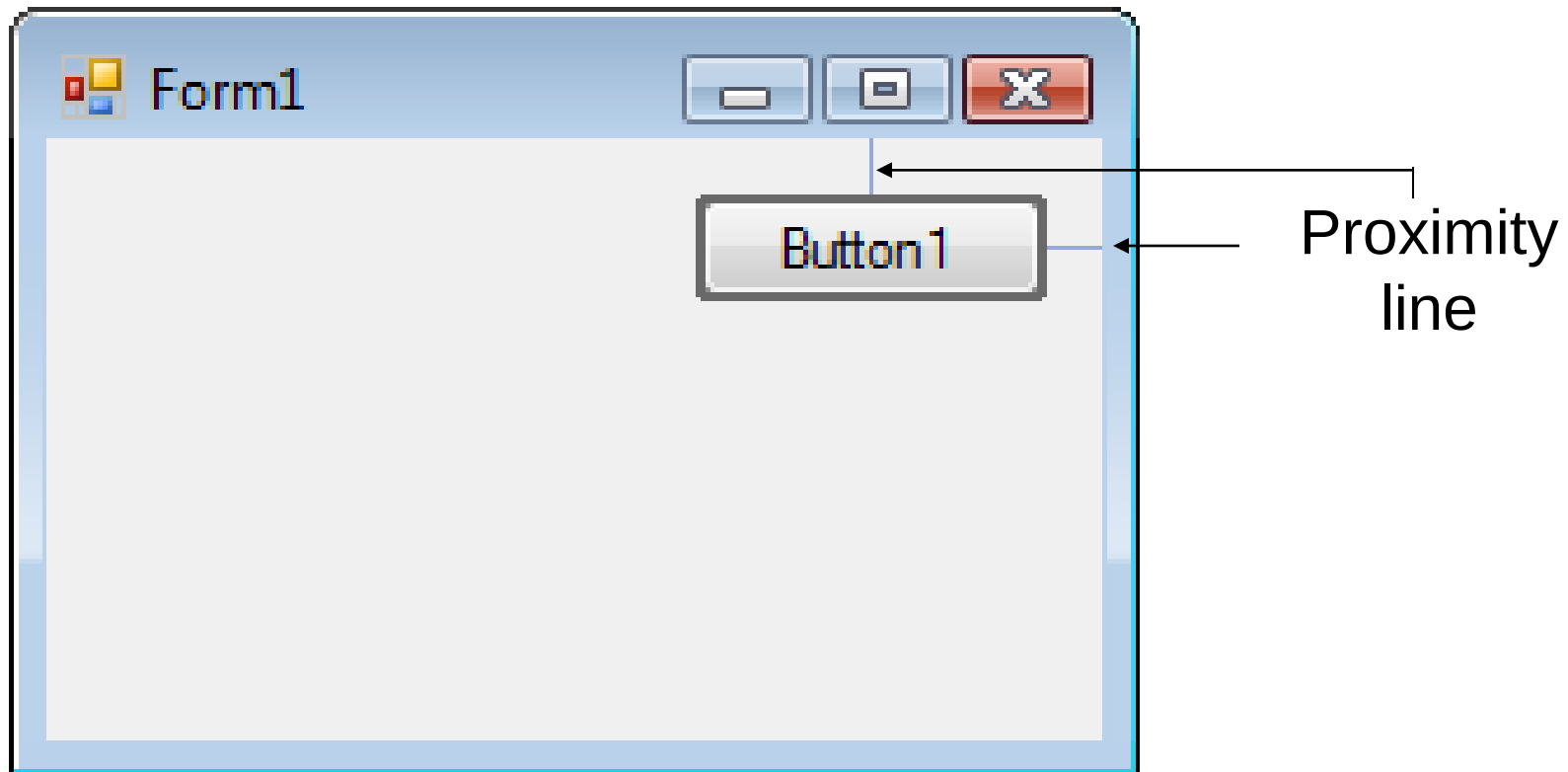
- Demo the previous
 - Label
 - Button
 - Colours
 - Font
 - Renaming
- Excellent walkthroughs in book

- Hides Toolbox when not in use
- Vertical push pin icon indicates auto hide is disabled.
- Click the push pin to make it horizontal and enable auto hide.

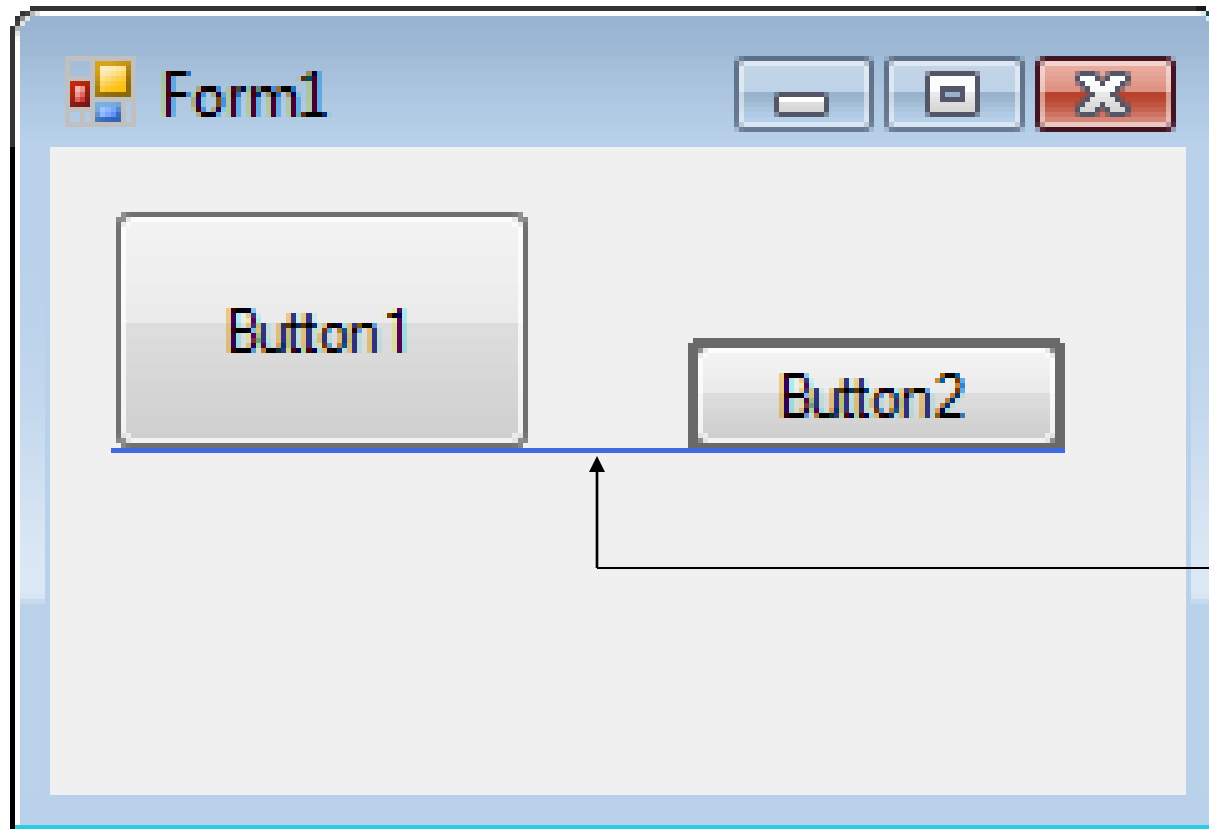


Push pin

POSITIONING CONTROLS

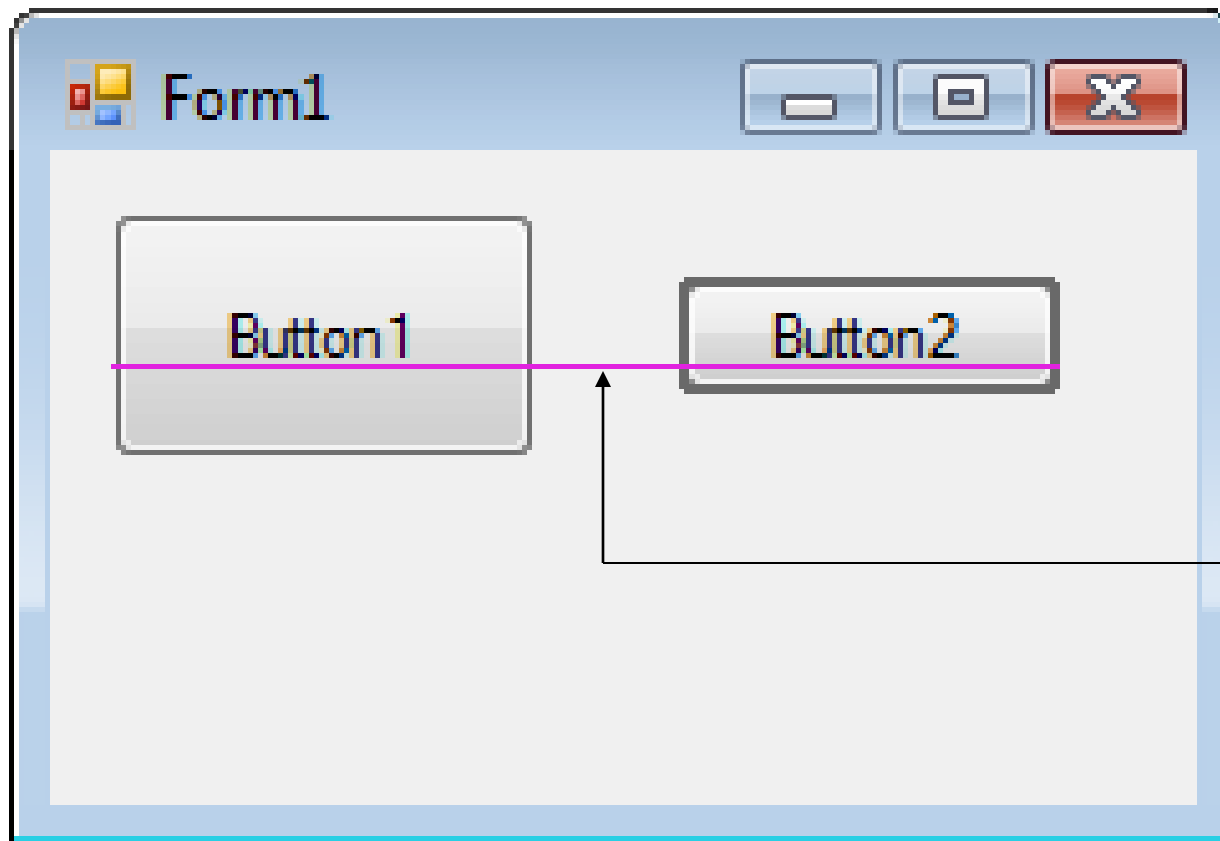


ALIGNING CONTROLS

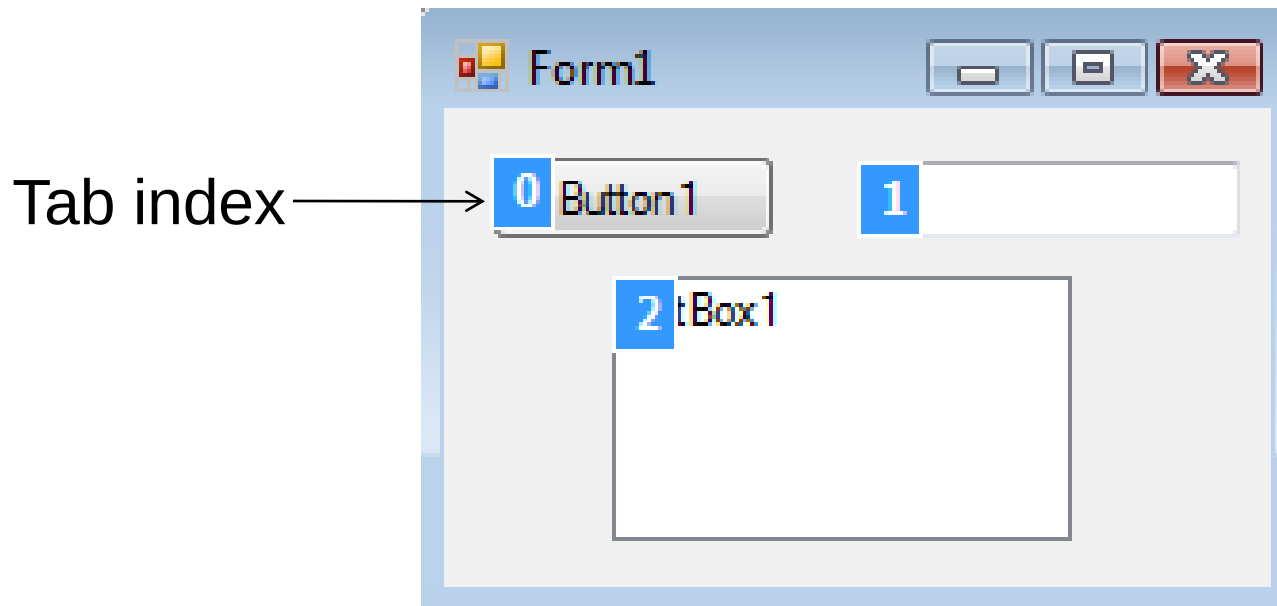


Snap line

ALIGNING CONTROLS



Snap line



The tab indices determine the order in which controls receive the focus during tabbing.

2.3 VISUAL BASIC EVENTS

- An Event Procedure Walkthrough
- Properties and Event Procedures of the Form
- The Header of an Event Procedure

- An **event** is an action, such as the user clicking on a button
- Usually, nothing happens in a Visual Basic program until the user does something and generates an event
- What happens is determined by statements

SAMPLE STATEMENTS

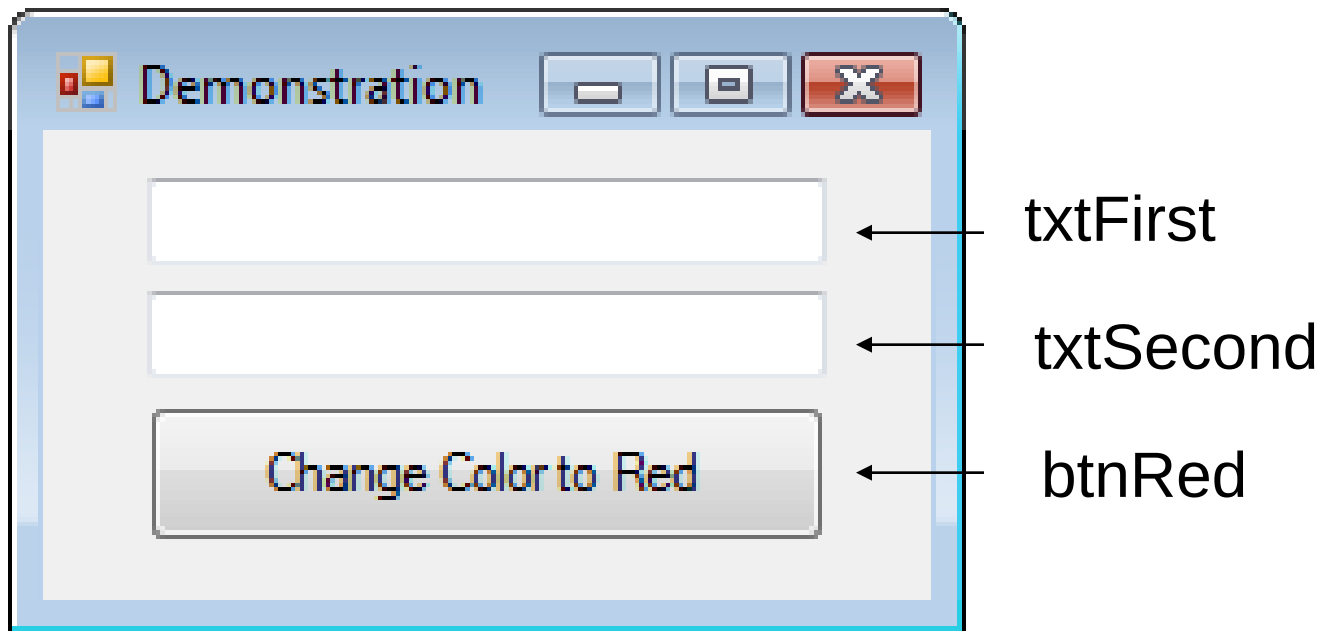
General Form:

controlName.property = setting

- `textBox.ForeColor = Color.Red`
- `textBox.Visible = True`
- `textBox.Text = "Hello World"`

What do the above mean?

SAMPLE FORM



What events could there be?

- When you click on a text box, a cursor appears in the text box, and you can type into the text box
- Such a text box is said to have the **focus**
- If you click on another text box, the first text box loses the focus and the second text box receives the focus

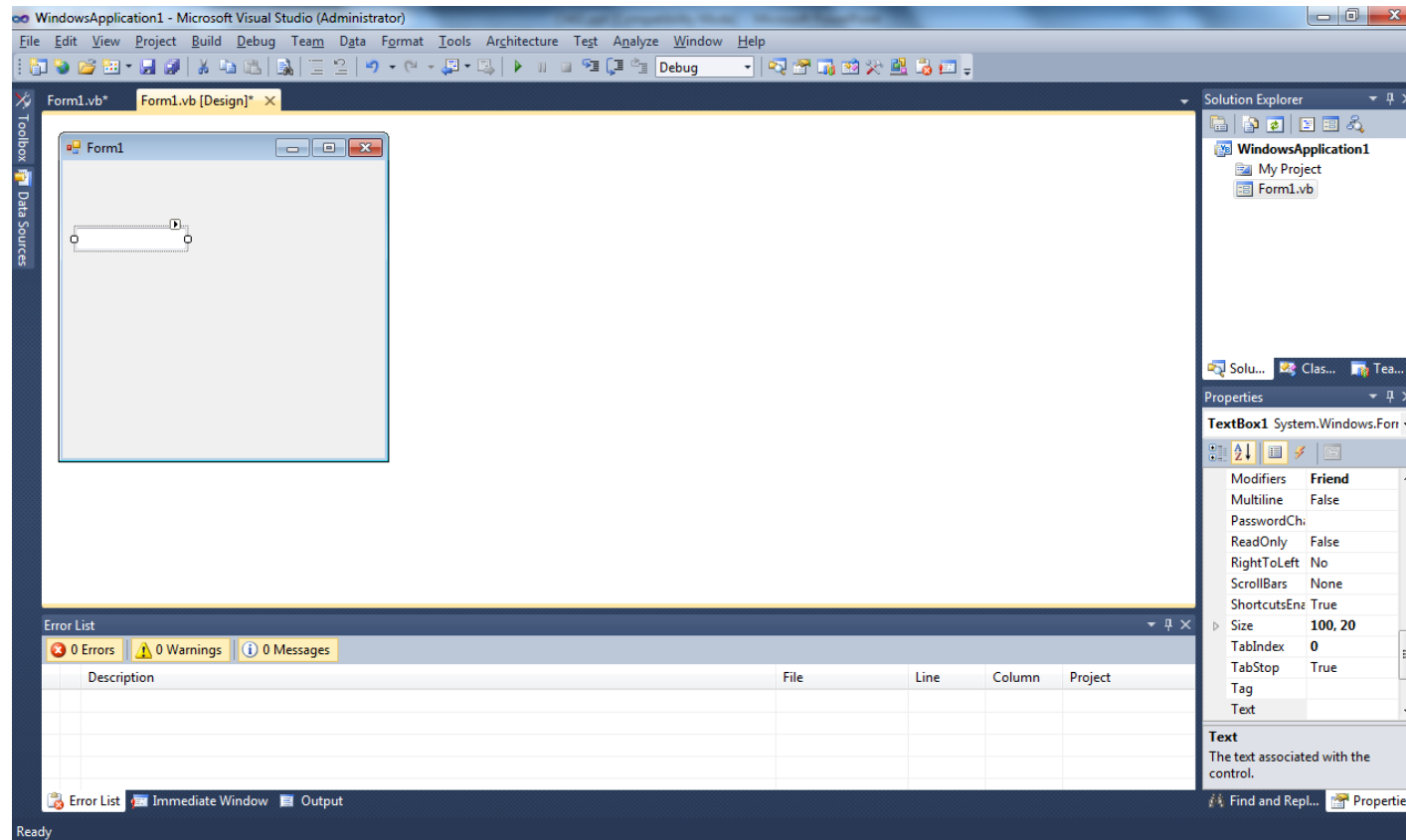
EXAMPLES OF EVENTS

General Form:

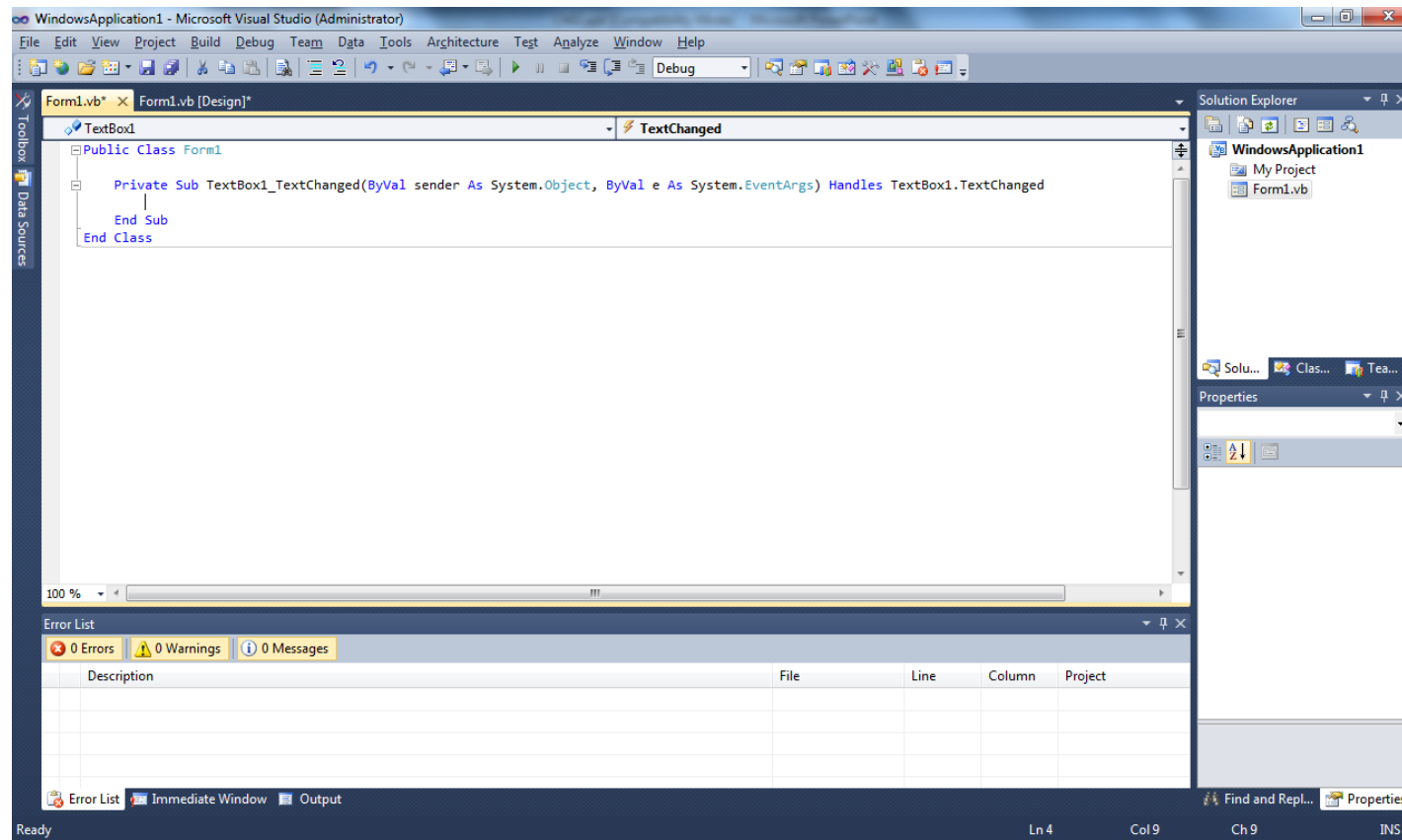
controlName.event

- btnShow.Click
- txtBox.TextChanged
- txtBox.Leave

- Take form, double click on textbox

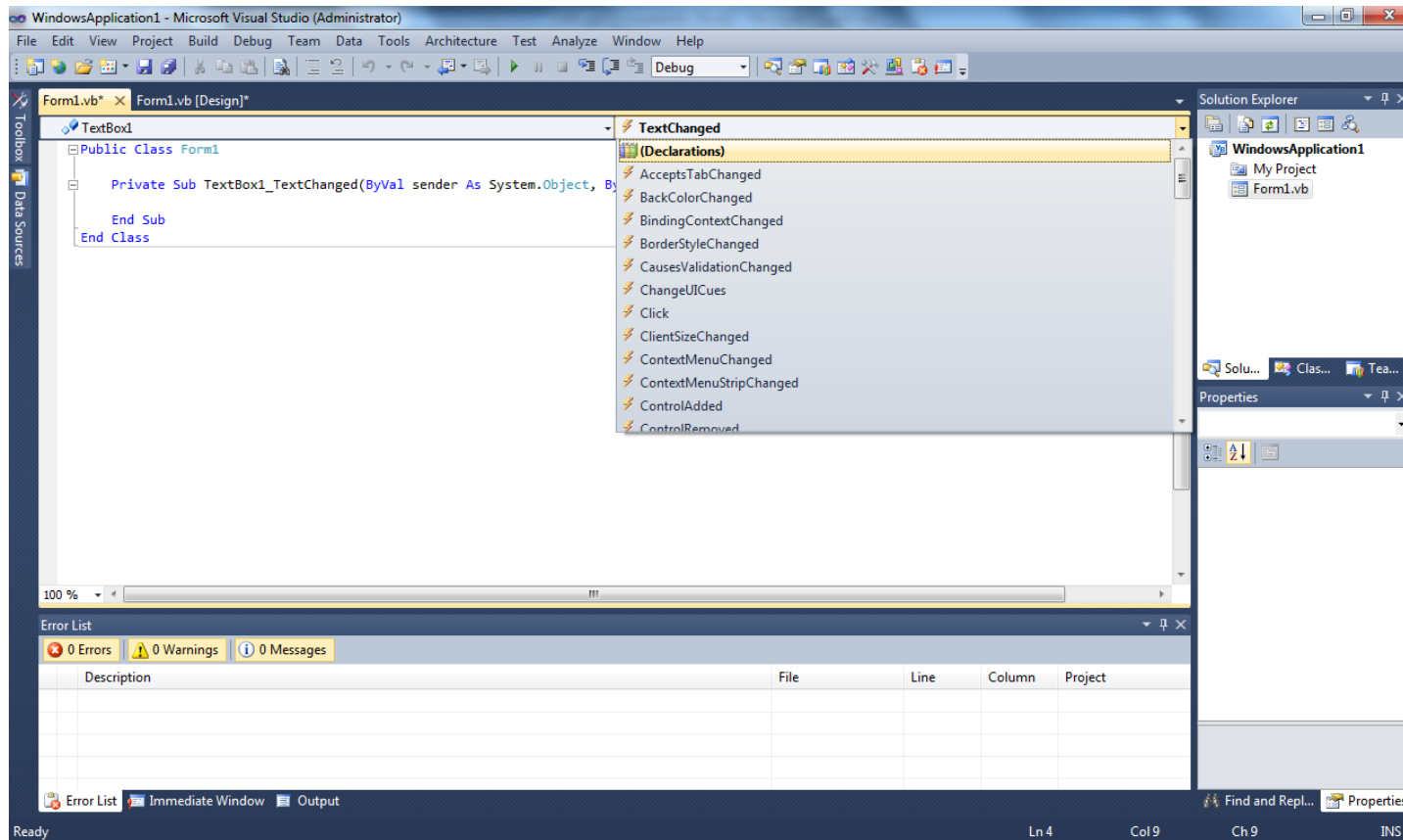


- Enter area to write code for control



EXAMPLES OF EVENTS

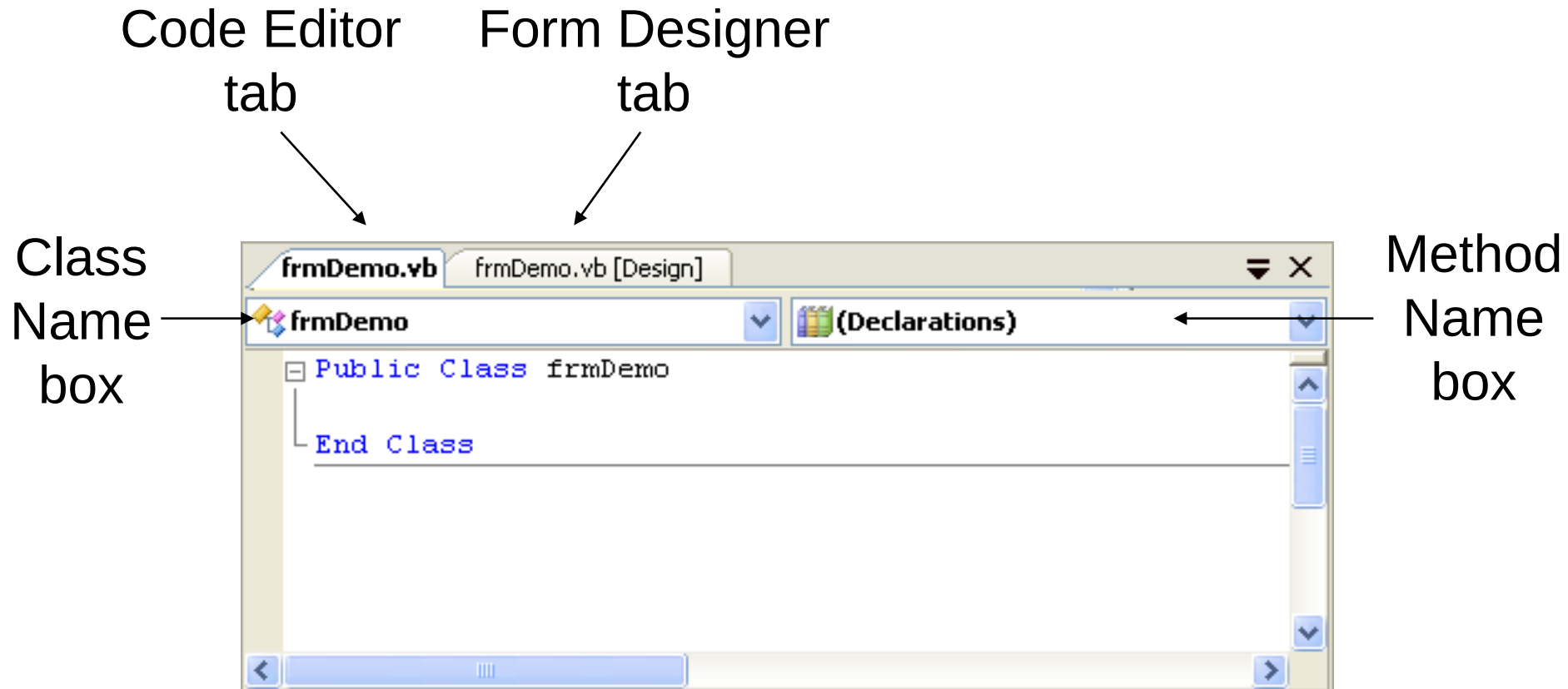
○ Select Event



3 STEPS IN CREATING A VB PROGRAM

1. Create the interface; that is, generate, position, and size the objects
2. Set properties; that is, configure the appearance of the objects
3. Write the code that executes when events occur

CODE EDITOR



STRUCTURE OF AN EVENT PROCEDURE

```
Header { Private Sub objectName_event(...)
        Handles objectName.event
        statements
      End Sub
```

(...) is filled automatically with
(ByVal sender As System.Object, ByVal
e As System.EventArgs)

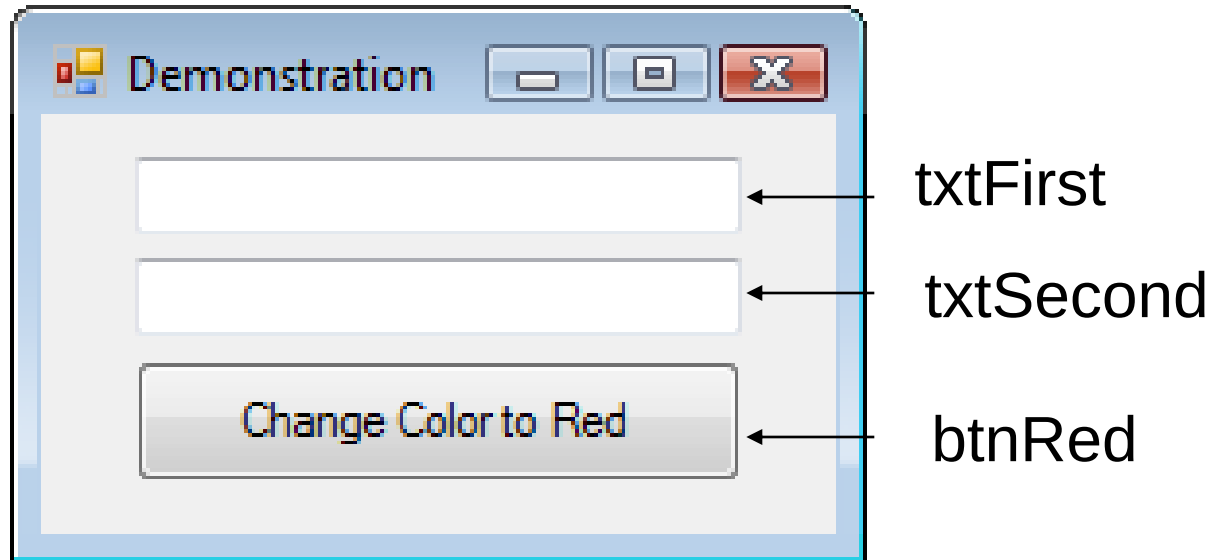
EVENT PROCEDURE

Create an Outline for an Event Procedure; i.e.
header and End Sub

- Double-click on a control
or
- Use the Class Name and Method
Name boxes.

(We nearly always use the first
method.)

SAMPLE FORM



Double Click on txtFirst

CODE FOR WALKTHROUGH

```
Public Class frmDemo
    Private Sub txtFirst_TextChanged(...)
        Handles txtFirst.TextChanged

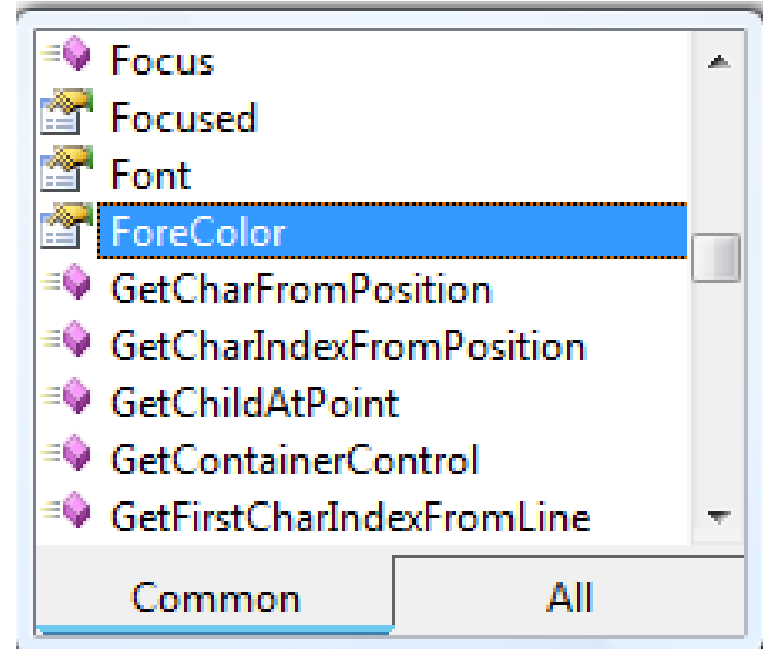
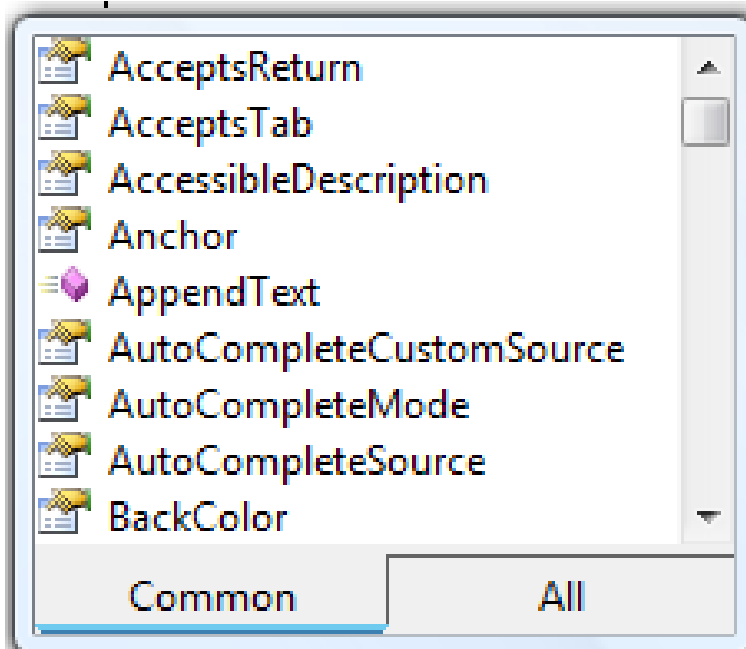
    End Sub
End Class
```

CODE FOR WALKTHROUGH

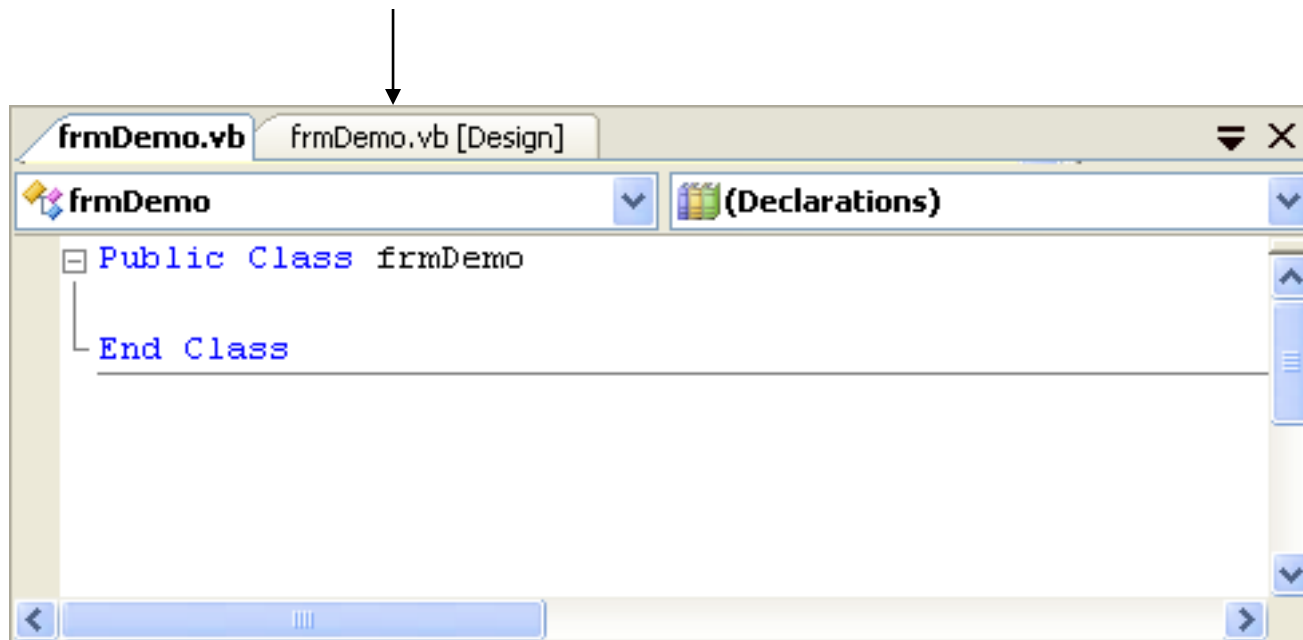
```
Public Class frmDemo
    Private Sub txtFirst_TextChanged(...)
        Handles txtFirst.TextChanged
            txtFirst.ForeColor = Color.Blue
        End Sub
    End Class
```

Automatically pops up to give the programmer help.

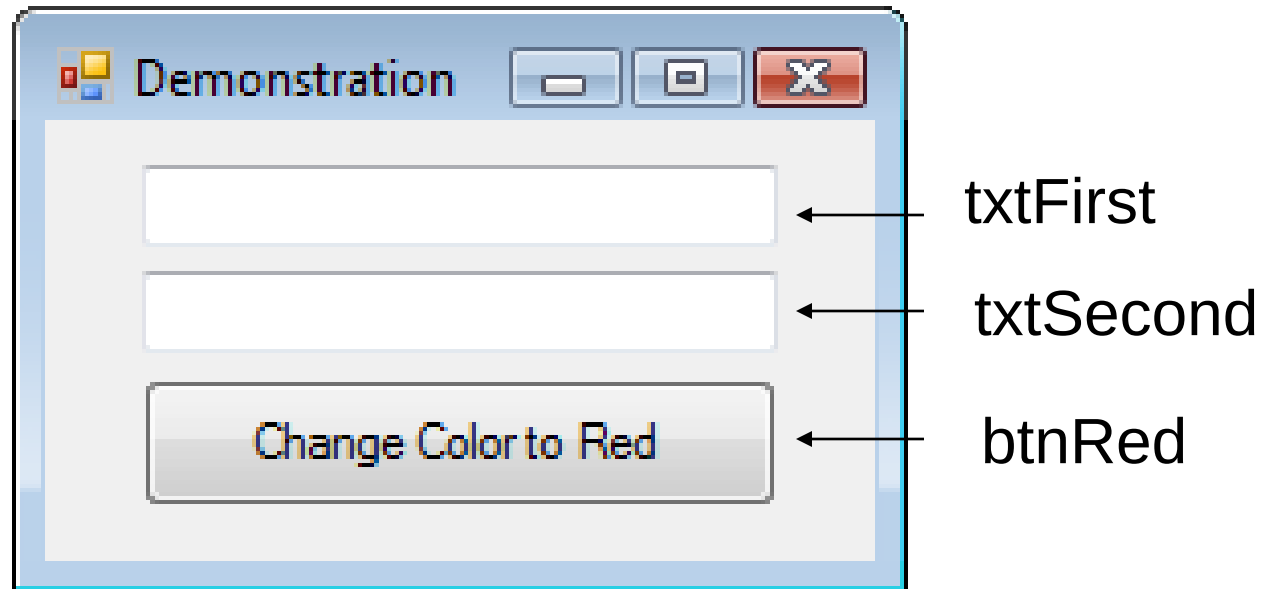
txtFirst.



Click tab to return to Form Designer



SAMPLE FORM



Double-click on btnRed

CODE FOR WALKTHROUGH

```
Public Class frmDemo
    Private Sub txtFirst_TextChanged(...)
        Handles txtFirst.TextChanged
        txtFirst.ForeColor = Color.Blue
    End Sub

    Private Sub btnRed_Click(...)
        Handles btnRed.Click

    End Sub
End Class
```

CODE FOR WALKTHROUGH

```
Public Class frmDemo
    Private Sub txtFirst_TextChanged(...)
        Handles txtFirst.TextChanged
            txtFirst.ForeColor = Color.Blue
    End Sub

    Private Sub btnRed_Click(...)
        Handles btnRed.Click
            txtFirst.ForeColor = Color.Red
    End Sub
End Class
```


EVENT PROCEDURE

- Select txtFirst from Class Name box drop-down list
- Select Leave from Method Name box drop-down list

CODE FOR WALKTHROUGH

```
Private Sub txtFirst_Leave(...) Handles txtFirst.Leave  
  
End Sub
```

CODE FOR WALKTHROUGH

```
Private Sub txtFirst_Leave(...) Handles txtFirst.Leave  
    txtFirst.ForeColor = Color.Black  
End Sub
```

HEADER OF EVENT PROCEDURE

`Private Sub btnRed_Click(...) Handles btnRed.Click`

↑
Name, can
be changed.

↑
Identifies event

`Private Sub Button_Press(...) Handles btnRed.Click`

HANDLING MULTIPLE EVENTS

Event procedure can be invoked by two events.

```
Private Sub Happening(...)  
    Handles btnRed.Click, txtSecond.Leave  
        txtFirst.ForeColor = Color.Red  
End Sub
```

ALTERING PROPERTIES

- The following won't work: **frmDemo.Text = "Demonstration"**
- The form is referred to by the keyword *Me*.
 Me.Text = "Demonstration"

TEXT FILE

- Holds data to be processed by programs
- Can be created, viewed, and managed by the Visual Basic IDE
 - Create: Use “Add New Item” input dialog box
 - View: Click on filename in Solution Explorer
 - Manage: Use Solution Explorer like Windows Explorer

SAMPLE TEXT FILE

Mike Jones

9.35

35

John Smith

10.75

33

PAYROLL.TXT

SAMPLE TEXT FILE

<i>name</i>		Mike Jones
<i>hourly wage</i>		9.35
<i>hours worked</i>		35
		John Smith
		10.75
		33

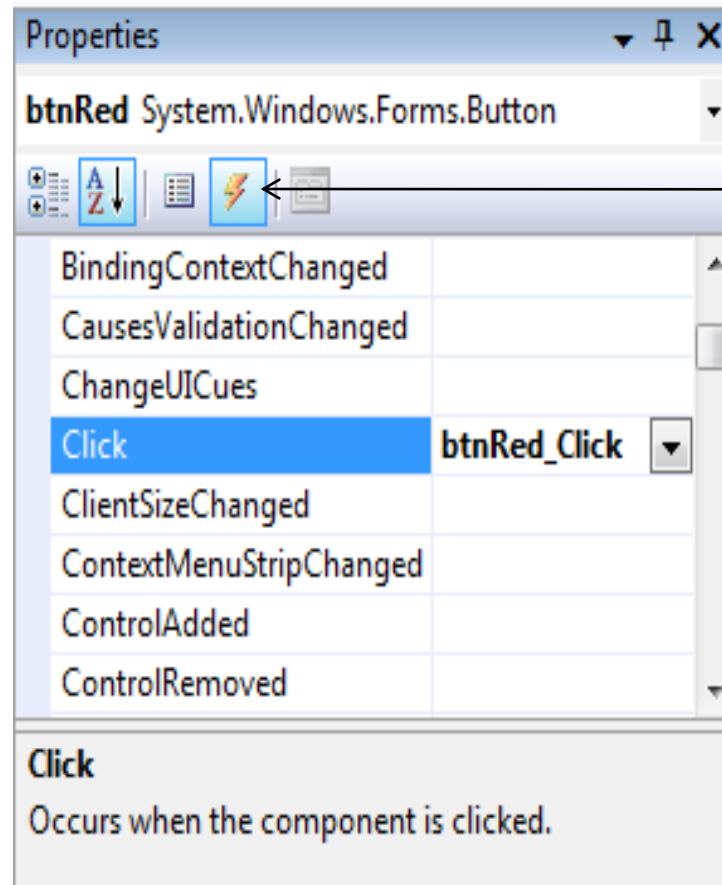
PAYROLL.TXT

OUR TEXT FILE CONVENTION

- PAYROLL.TXT uses one item of data per line.
That will be our convention in this book.

POSSIBLE EVENTS

Use Property Window to View all Possible Events
for a Control



Events
button

RUN AN EXISTING PROGRAM

1. Click on Open Project in the File menu
2. Navigate to the program's folder
3. Click on the program's folder
4. Double-click on the file with extension .sln
5. In the Solution Explorer double-click on the file with extension .vb. (The Form Designer will appear.)
6. Press F5 to run the program

RUN PROGRAM

