**UNIT -1**

Introduction to Visual Basic: Integrated Development Environment (IDE) features – Working with form properties – setting form‟s properties - introducing form events and form methods. Variables in Visual Basic: Declaring Variables – The Scope of a variable – Module level variables – Constants – Creating your own constants – Scope of a constant – converting data types – Arrays – Declaring arrays – Fixed size arrays – Dynamic arrays – Preserve keyword – Redim.

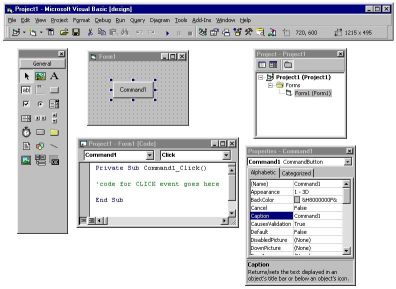
**INTRODUCTION TO VISUAL BASIC**

* Visual basic is an event driven programming language.
* Visual->It refers to a method used to create the graphical user interface(GUI)
* Basic->Visual basic has evolved from the BASIC (Beginner's All-purpose Symbolic Instruction Code) language.
* It is an ideal medium for developing Windows based application.
* It provides a complete set of tools to develop an application

**INTEGRATED DEVELOPMENT ENVIRONMENT**

The windows that are displayed when the user start VB are collectively known as the Visual Basic Integrated Development Environment (IDE).

**IDE IN VB**

[](http://www.garybeene.com/images/t-vb6.jpg)

**Features ofIntegrated Development Environment**

IDE has the following features

1) Project Explorer

2) Properties Window

3) Menu bar

4) Tool bars

5) Tool box

6) Object Browser

7) Form Designer

8) Code Editor

9) Form Layout window

**1) Project Explorer**

It organizes the application as one project. By default it located on the right side of the window. All the controls and codes that are used in the applications are stored in separate files. It is called as a ‘Project Explorer’.

It has 3 icons on its tool bar



Icon to view the code



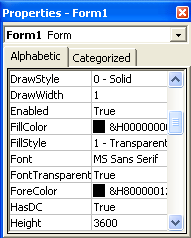
Icon to view the controls



Icon to show/hide the forms

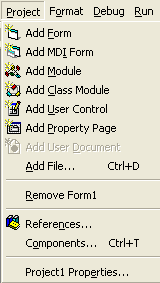
**2) Properties Window**

* This window lists all the properties for an object or control used in VB.
* The user can change the properties like the caption, the height, the width etc.



**3)Menu bar**

* It displays the commands use to work with Visual Basic.
* There are menus like File,Edit,View,Window,Project,Format,Debug,etc



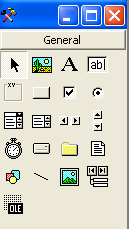
**4) Tool bars**

It provide quick access to commonly used commands in the programming environment



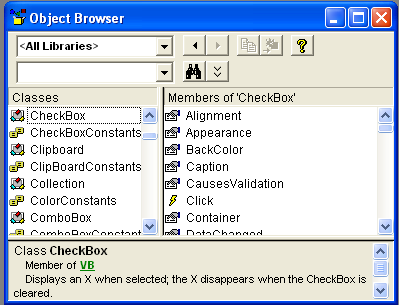
**5)Tool box**

It provides a set of tools that helps at design time to place controls on a form.



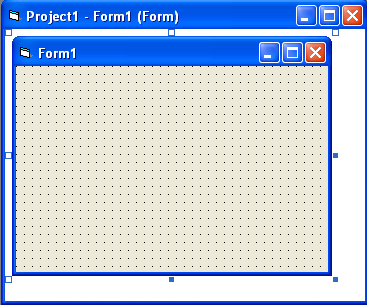
**6)Object Browser**

It lists the objects available for use in the project. It also helps to see what methods and properties are available for those objects.



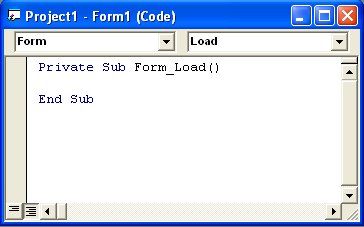
**7) Form Designer**

* It serves as a window that helps to design the interface of your application.
* Users can add controls, graphics and pictures to a form.
* Each form has its own form designer window



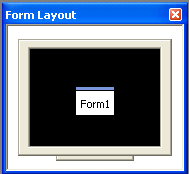
**8) Code Editor**

* It serves as an editor for entering the application code.
* A separate code editor window is created for each form in our application.
* The programmer can change the font size,caption,font style of a text through the code also.



**8)Form Layout window**

Form Layout window allows us to position the forms in our application using a small graphical representation of the screen.



**WORKING WITH FORM PROPERTIES**

A form in Visual Basic has as many as 50 properties. Each of these properties will affect the appearance and behavior of the form.

**Working with Properties Window**

We can assign the properties to a control during the Design Time by using Properties Window. It allows us to change the current settings of the controls.

Steps to access the properties

1. Click the right mouse button
2. Choose the Properties option from the pop-up menu

We can also select the control and press **F4 .**It displays the Properties window for that control

**SETTING FORM PROPERTIES**

To open a Properties window Click on the form Press F4

It opens with the current setting for the form.

**Font**

Font Properties are

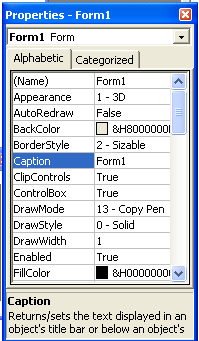
Font Name 🡪Name of the Font

Font Bold 🡪 If set to True, the text will be displayed in bold

Font Size 🡪 To set the size of the text in points

To access the font property, we can scroll up/down till we reach that property.

Or press Ctrl+Shift+First Letter of property, in this case press Ctrl+Shift+F.

****

**Name**

* It refers to the name of the form.
* In VB the default name of the forms are Form1, Form2, Form3, etc.
* We have to give the meaningful names for the forms. Forexample, the form for Invoice program should be name as frmInvoice.
* The startup form is named as frmMain
* The name of a form cannot be changed at runtime.

**Caption**

* This is the name that appears on the Title bar of the Form.
* This can be changed at runtime.
* The caption must be meaningful and informative to the user.

**Picture**

* The name of the file that contains the picture to be displayed on the form.
* This can be changed at runtime.
* Every time that the user changes from one module to another, the picture can be changed.

**Background Color**

* This property determines the background color of the Form.
* To change the settings, double click on the value on the right.
* A color palette will pop up. Select the color of our choice

**The Control Box**

* The value of this property is True or False.
* If it is true, the Control Box is visible on the top left hand corner of the form.
* If it is false, the Control Box is not visible

**Min and Max Button**

* The value of these properties is True or False.
* If it is true, the user can minimize or maximize the form during runtime.
* If it is false, the user can’t do this access.

**Movable**

* The value of this property is True or False.
* If it is true, the user can move the form.
* If it is false, the user can’t move the form during runtime.

**Border Style**

* This property determines the type of window that displays at runtime.
* It allows the user to resize the window or allow to move the window.
* It is advisable not to allow the user to resize the window because important application interface might get hidden.

**INTRODUCING FORM EVENTS AND FORM METHODS**

**Form Events**

1)Initialize

2)Load

3)Resize

4)Activate

5)Deactivate

6)QueryUnload

7)Unload

8)Terminate

**1) Initialize**

The Initialize event is the first event of a form when the program runs.

This event is to initialize the form’s properties.

**Example:**

Private Sub Form\_Initialize()

Text1.Text = ""

Text2.Text = ""

End Sub

**2) Load**

During the Load event, the Form with all its properties and variables is loaded in memory

**3) Resize**

After the Load event, the form receives the Resize event. The form's Resize event is also raised when you resize the form either manually or programmatically.

**4)Activate**

After the Resize event, activate event fires.

It occurs when the Form gets user input

It also occurs when the Show method or Set Focus method of the form is called

**5)Deactivate**

This event occurs when another form gets the focus.

**6)QueryUnload**

When you close or unload the form, the form first receives the QueryUnload event.

It allows the option to abort the Unload event

**7) Unload**

When the user closes the Form, the Form is unloaded from the memory.

We can use this event to warn the user that data needs to be saved before closing the application

**Example:** In this program, you cannot close the Form keeping the text field blank

Private Sub Form\_Unload(Cancel As Integer)

If Text1.Text = "" Then

MsgBox "You cannot exit keeping the text field blank"

Cancel = True

End If

End Sub

When Cancel = True, you cannot close the Form. The Cancel parameter is used to cancel the the form's unload operation.

**8)Terminate**

The final event in the lifecycle of a Form.

All the memory that was held for the Form variables is released

**Form Methods**

Forms have methods that are used in code to change the appearance and behaviour of the form at runtime.

1)Move Method

2)Graphic Methods

3)Show Method

4)Hide Method

**1)Move Method**

It allows the programmer to position the form at a desired location on the screen.

**Syntax**

FormName.Move Left,[Top],[Width],[Height]

**Example**

FrmInvoice.Move 0,0,3500,4000

**2)Graphic Methods**

There are other properties for drawing lines,circles,clearing the form for any drawing object.

There are methods like getting the color of paint at a particular location.

Circle : To draw a circle

Line : To draw a line

Pset: To draw a point with a given color at a given location

Point: Returns the color of screen at the given location

**3)Show Method**

The Show method displays the form to the user.

It is like setting the ‘Visible’ property to True.

The Load event will not display the Form

**Example:**

Load frmInvoice 🡪It load the form but not show it

frmInvoice.Show🡪 It shows the form

It has 2 optional parameters

FormName.[showStyle],[Ownerform]

Show Style 🡪 Determines the nature of the form

Owner Form 🡪 Determines the form that is calling the new form

**Show Style**

There are 2 options,

1)The Form can be opened as a **Modal**Form. The user will not be able to interact with another form as long as this form is active

2) The default option is a **Modeless**form. The user can interact with other Forms even while this Form is displayed.

**4)Hide Method**

It makes the form invisible to the user. It is the opposite of the Show method.

**Syntax:**

frmInvoice.Hide

**VARIABLES IN VISUAL BASIC**

**Variables**

The various values used during computation are stored in a element is called a variable.

**For example**, we can have a variable called ‘ Total Bill Amount’ that will hold the total invoice value, a variable called ‘Item Rate’ that will hold the rate of the item.

**DECLARING VARIABLES**

Before the variable is used we need to inform the program that this particular word is a variable. The process or methods of providing this information is called **declaring** a variable.

Variables in VB are declared using the **DIM** statement

**Syntax :**

**Dim**variablename [**As** type]

**For example :**

**Dim** Total Bill Amount [**As** integer]

**Rules for naming a variable:**

* Must begin with an alphabet
* Must not have an embedded period or special character
* Must not exceed 255 characters
* Must be unique within the same scopes

**Data Types**

* The data type of a variable determines how the bits/bytes representing the values are stored in the computer’s memory.
* Variables have a name and a data type
* All variables have a data type that determines what kind of data they can store.

**Example:**

In the Invoice program, the variable Item Name hold the values like brush, tooth paste etc. which are purely strings.

The following table explains the basic data types and its sizes

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Type of Values** | **Size** |
| Integer | Numeric Values | -32,768 to 32,767 |
| Long (Long Integer) | A Wider range of Integers than Integer | -2,147,483,648  To  2,147,483,647 |
| Single | Numbers with decimal places | -3.402823E38 to  -1.401298E-45 for Negative Values  1.401298E-45 to  3.402823E38 for Positive values |
| Double | Wider range of Integers than Single | 1.79769313486232E308  To  -4.94065645841247E-324 for negative values  4.94065645841247E-324 to 1.79769313486232E308 |
| Currency | Monetary Values | -922,337,203,685,477.5808  To 922,337,203,685,477.5807 |
| String | Text or String Values | 0 to approx. 2 billion for variable length. 1 to approx.65,400 for fixed length. |
| Byte | Less than the value 255 | 0 to 255 |
| Boolean | True or False values |  |
| Date | Date values inclusive of and between  Jan 1, 100 to Dec 31, 9999 |  |
| Object | Reference to objects in visual basic and other applications. |  |
| User Defined  (Using Type) | The range of each element is the same as the range of its data type |  |
| Variant | General Purpose variable holds most other types of variables values (with number). Any numeric value upto the range of a double. With Character values, it has the same range as for a variable-length string. |  |

**Integers**

This datatype is used to store whole numbers.

It occupies only 2 bytes of memory.

Quick fast for doing calculations.

**Long**

It occupies twice as much as the Integer.

It is used only the calculations involve very large numbers.

Slower than Integers.

**Single**

It is used to store fractions.

It occupies 4 bytes of memory.

**Double**

It occupies 8 bytes of memory.

It is used when the users need accuracy in calculations

**Currency**

It is used to hold the values related to item rates, payroll details and other financial functions

**Byte**

It occupies 1 byte of memory.

It cannot hold negative numbers.

**Boolean**

Default value for this type is zero.

Zero value represents False and a Non-Zero represents True.

**Date**

It occupies 8 bytes of memory.

It is displayed as per settings in the computer.

**The String Data Type**

It is the most commonly used datatype.

We can declare the variable in 2 ways

Variable-length string

Fixed-length string

To remove the trailing spaces use Trim and Rtrim functions.

**Example**

Dim ItemNameAs String \* 30 🡪 Fixed –length string

Dim ItemDescriptionAs String🡪 Variable length string

**Object Data Type**

In VB, forms,controls,procedures and recordsets are all considered as Objects

It occupies 4 bytes of storage.

A variable declared as an Object that can immediately assigned using Set statement.

**Example**

Dim objDbAs database

Set objDb = OpenDatabase(“C:\DATA\ERP.mdb”)

**Variant Data Type**

If we didn’t mention the data type, the variable is given the Variant datatype by default.

Conversions can do automatically.

A variant datatype is a variable that can change its type freely.

It can accept text, numeric data or byte data.

**IsNumeric** function performs the checking of whether variant variable contains a value of type number

**Example**

Dim VarValue🡪 Variant by default

VarValue = “100” 🡪 Holds the String

VarValue = “Valluvar” 🡪 Holds the String

It can contain values like

The Null Value

The Empty Value

The Error Value

**The Null Value**

It is used in database applications to indicate unknown or missing data. If you assign Null to a variable other than Variant type causes an error

Assign Null with Null Keyword : Z= Null

IsNull function is to test, if a Variant variable contains Null

Ex: If IsNull(X) Then

Debug.print …..

End If

**The Empty Value**

A Variant variable has the Empty value before it is assigned a value

The Empty value is different from 0, a zero-length string(“”) or Null value.

IsEmpty function is to test for the Empty value

Ex: If IsEmpty(X) Then ……….

Empty value can be assigned using the **Empty** keyword

**The Error Value**

In a variant, Error is a special value used to indicate that an error condition has occurred in a procedure.

An error value is created by converting a real number using the **CVErr** function.

**THE SCOPE OF A VARIABLE**

* Code is written for each control and individually for each event of that control, while variables are declared for each event or procedure.
* The scope of a variable is the range from which the variable can be referenced a procedure, a form, and so on.
* The value of a variable in one procedure cannot be accessed from another procedure. The value of a variable is **local** to that procedure.
* Variables declared with the **Dim** statement within a procedure exist only as long as the procedure is executing. When the procedure finishes, the value of the variable disappears.
* These characteristics allow you to use the same variable names in different procedures without confusion.
* The scope of a variable is determined by the way it has been declared. It can be
  + - Procedure level variable
    - Module level variable

**Procedure level variable**

Procedure level variables are recognized only in the procedure in which they are declared.It is also known as local variables.

Values in local variable declared with Static exist the entire time of our application running.

Syntax: Static intCounter As Integer

The advantage of this method is that this variable can be accessed only by this procedure and not by other procedure.

**Module Level Variables**

Variables declared as module level will be available to all procedures within that module.

However they will not be available to procedures in other modules.

This variable is declared using the Private keyword, in the declaration section of the module.

Ex: Private intcount As Integer

To make the variable globally use the Public keyword.

Public variables cannot declare in the procedure. They can be declared only on the declaration section of the module.

Ex: Public intTemp As Integer

**Scope of a variable:**

|  |  |  |
| --- | --- | --- |
| **Scope** | **Private** | **Public** |
| Procedure – Level | Variables are private to the procedure in which they appear. | Not applicable. You cannot declare public variables within a procedure. |
| Module – Level | Variables are private to the module in which they appear. | Variables are available to all modules. |

**Public and Private Variables**

If we have a public and private variable with the same name, the reference method will be change.

**Ex:**

Public X as Integer

X=15 🡪The assignment is done later

Dim X as Integer

X=10 🡪 Accessing the variable X in the procedure

If we have to access the variable X in the module level.:

Debug.print.Module1.X

This will display the value of 15

Local variable will always be accessed in preference to the public variable. This is called **Shadowing.**

**Declaring Variable**

Declaring a variable using the Public keyword, it will be available throughout the application.

Declaring a procedure level variable using the Static keyword preserves its value even when a procedure ends.

**Implicit Declaration**

We can use the variables without declaring it.

VB automatically creates a variable with that name is called as Implicit Declaration

**Option Explicit**

It will avoid the problem of misnaming variables.

Place the statement **‘Option Explicit’** in the declaration section of the Form,Module or Class.

Another way to insert ‘**Option Explicit’** is

From the tools menu 🡪Choose options🡪 Click the Editor tab🡪Check the Require Variable declaration

**CONSTANTS**

Constants store values like variables,but those values remain constants throughout the execution of an application.

**Example:**

The value ‘Pi’ can be represented as 3.14.This number assigned to a ‘Constant’ and can used in all calculations.

**CREATING YOUR OWN CONSTANT**

Syntax:

[Public | Private] Constconstantname [As type]=expression

Constant name should be a valid name,

Type is the datatype,

Expression is the numeric or string value that has to assigned to the constant .

**SCOPE OF A CONSTANT**

By declaring a constant using the Public keyword, it is available throughout the application.

Declaring a constant in a procedure will be available to that procedure only.

**Circular Constant**

Constants can be defined with reference to other constants.

**Ex:** Public ConstconA=ConB \* 1.414

Both constants are available throughout the application. Visual Basic generated an error.

This method of defining constants where each defines in terms of the other is called **Circular reference.**

**CONVERTING DATA TYPES**

To convert the text data into numeric, convert numeric data to currency data type, VB provides functions to convert values into data type.

|  |  |
| --- | --- |
| **Conversion function** | **Converts an expression to** |
| Cbool | Boolean |
| Cbyte | Byte |
| Ccur | Currency |
| Cdate | Date |
| CDbl | Double |
| Cint | Integer |
| CLng | Long |
| CSng | Single |
| CStr | String |
| Cvar | Variant |
| CVErr | Error |

**ARRAYS**

An array is a set of similar items. All items in an array have the same name and are identified by an index.

**Syntax**

Dim Varname [([subscripts])] as [New] type [,varname…..]

**DECLARING ARRAYS**

Dim nums(10) as integer

Dim x(10 to 20 ) as integer.

In the first case, ‘nums’ is a set of 11 integers.

Nums(0) is the first element of the array ‘nums’

Nums[10] is the eleventh and the last element of the array ‘x’

It can be classified into 3 types,

* + Fixed-Sized array
  + Dynamic arrays and
  + Multidimensional arrays

**FIXED SIZE ARRAYS**

The scope of the array will depend upon the method of declaration

1. To create a local array, use the Private statement in a procedure to declare the array.

Dim Countries(10) As Integer

1. To create a module-level entry, use the Private statement in the declarations section of a module to declare the array.

Private Counters(10) As Integers

1. To create a public array, use the Public statement in the Declarations section of a Form.

Public Counters(10) As Integers

In the case of fixed sized arrays, it is compulsory to enter the upper bound of the array.It is the upper limit for the size of the array

Private Counters(10) As Integers

Will result in an array of 11 elememts

LBound function🡪 Returns the lower bound of the array

UBound function 🡪 Returns the upper bound of an array.

**DYNAMIC ARRAYS**

A dynamic array can be resized at any time and this helps you to manage memory efficiently.

The ReDim is used in conjunction with the Dim statement while declaring these arrays.

Declaring a dynamic array

1. You declare the array as dynamic by giving it an empty dimension list.

Dim DynSaleval ( )

1. Use the ReDim statement to allocate the actual number of elements.

ReDimDynSaleval( 11, 4 )

Note : ReDim is an executable statement and can appear only in a procedure. Unlike the Dim and Static statements, it makes the application carry out an action at run time. You can use ReDim each time you want to change the upper and lower bounds of a dimension. You cannot however change the number of dimensions.

**Example:**

Dim saleval() As Integer

Redimsaleval(12,6)

//creates a 2 dimensional array

The values for the subscripts of the array can be passed using variables

**Example:**

Dim saleval() As Integer

Dim Months As Integer

Dim DeptsAs Integer

Months =12

Depts =6

ReDimsaleval(Months,Depts)

**PRESERVE KEYWORD**

To expand or increase the size of the array the Preserve keyword is used.

Ex: ReDim Preserve Saleval(12,9)

It will not destroy the data that has already been entered.

It adds the more columns to the table. We can enlarge an array by one element without losing the values of the existing elements use the UBound function.

Ex: ReDim Preserve DynSaleval(UBound(DynSaleval)+1)

**UNIT – 2**

Writing Code in Visual Basic: Language constructs – For… Next, The while loop, Select Case… End Select, Exit statement, With Structure. Introduction to standard controls – Command buttons – Text boxes – Labels – Option buttons – Check boxes – Frame controls – List boxes – Combo boxes – Image objects – Picture boxes – Timer – Scroll bars – File system controls (Drive, DirList, File List boxes).

**WRITING CODE IN VISUAL BASIC**

VBA is the name of the programming language used in Visual Basic. It is the language used for the other desktop products from the Microsoft. It provides the code window to compose (view,write,edit) the VBA code.

**Opening the Code Window**

* Open the code window by pressing the right mouse button on the control. Choose the “view code” option.
* OR press F7 after clicking on the control.
* OR Click on View in the menu bar and click view code.
* OR right click the name of the form from the project explorer and select view code.
* The code window will open on the default event of that control. For example if the form has the focus, the code window will open on the form’s Load event.

**Parts of the Code Window**

**Project Name**

The name of the project to which the procedure belongs is displayed in the title bar.

**Module Name**

A project can contain more than one module. A form is also a module.

**Object Box**

When you open the code window with F7 it displays the name of the control selected. Click on the down arrow to display the names of all the controls associated with the form.

**Procedures/Events Box**

This will list all the events recognized by visual basic for the control or form displayed in the object box.

**Split Bar**

This will split the code window into two horizontal panes. To implement this, click on ‘windows’ on the menu bar. From the drop down menu select ‘Split’.

**Margin Indicator Bar**

The gray area on the left side of the code window is the margin indicator.

**Procedure View Icon**

It displays the selected procedure. Only one procedure at a time is displayed in the code window.

**Full Module View Icon**

It displays the entire code in the module

**The Procedure Separator**

It is a horizontal gray line that separate two procedures. It can be turned on or off.

**Structure of a Procedure**

The procedure is the piece of code that gets executed when the control that it is associated with senses an event. The events are mouse click, mouse move, or a method invoked by the code. It consists of,

* **Name**

Every procedure must have a name.We have a procedure called cmdExit\_Click()

* **Declaration area**

One should make all the declaration for variables, constants etc

* **Statements**

The procedure deals with basic execution. A procedure can have as many statements, but keeping that short and simple that helps to debug easy.

* **Terminator**

Every procedure is terminated by the’ End Sub’ statement

**LANGUAGE CONSTRUCTS**

**The For….Next Statement**

This structure is used when you want to execute a statement or a block of statements a certain number of times.

START

PROCESS

INCREMENT

SET UPPERLIMIT

IS UPPERLIM

CONTINUE

END

**Example:**

For int I=1 to 10

intTotal=intTotal+ intI

Next intI

* This loop will compute the sum of the numbers from 1 to 10
* The statement intTotal=intTotal+1 performed 10 times.

**The While Loop**

This structure executes a series of statements as long as a given condition is true.

**Syntax**

While condition

[statements]

Wend

**Example Code**

Dim intI as Integer

intI=1

While intI<10

intsum = intsum +1

intI = intI +1

Wend

**How does it works**

* + While checks the condition. If it is true the statements are executed.
  + When the program encounters the Wend statement, the control returns to while
  + If the condition returns true, the process is repeated.
  + If the condition is false, statements following the Wend are executed.

**Select Case….. End Select**

This method is used when the program has to execute one of several groups of statements, depending on the value of an expression.

Syntax

Select case testexpression

[Case expressionlist-n]

[statements-n]]..

[Case Else

[elsestatement]]

End Select

**Example code**

Select Case CandidateAge

Case <= 18

MsgBox you are in the Juniors

Case <= 50

MsgBox you are in the Seniors

Case Else

MsgBox you are in the Oldies

End Select

**How does it works**

* The test expression is evaluated with the case expression.
* If the two match, the statements following that case, upto the next Case or End Select in the case of the clause are executed.
* If the two do not match, the test expression is evaluated with the next case expression.
* If more than one case expression matches with the test expression, the statements of the first case only are executed.
* The case else is used to indicate the statements to be executed if no match is found between the test expression and the case expression.
* If there are no matches, program execution continues with the statements after the end select.
* The select case can evaluate any number of conditions without making the program difficult to read and debug.
* The Case Expression can evaluate ranges too.

Example: Case 1 to 10

Example: Case 10,15,20..

Example: Case Is<=10, where the Is keyword is used.

**Exit Statement**

This statement exits a procedure or block and transfers control immediately to the statement following the procedure call or the block definition.

Syntax

Exit {Do | For}

**Description**

**Exit Do**

* Immediately exits the Do loop in which it appears. Execution continues with the statement following the Loopstatement.
* Exit Do can be used only inside a Do loop. When used within nested Do loops, Exit Do exits the innermost loop and transfers control to the next higher level of nesting.
* **Example code**

Do\_While loop terminated using Exit\_Do statement.

i=1

Do while i<100

Text1.Text=Str(i)

I=i+2

If i>500 then

Exit Do

End if

**Exit For**

* Immediately exits the For loop in which it appears. Execution continues with the statement following the Nextstatement.
* Exit For can be used only inside a For...Next orFor Each...Next loop. When used within nested For loops,Exit For exits the innermost loop and transfers control to the next higher level of nesting.
* **Example code**

For i=1 to 100

Text1. Text=Str(i)

If i=50 then

Exit For

End If

Next

* This procedure code increments the value 1 by 1. If it reaches the condition i= 50, the Exit For statement is executed and it terminates the For\_Next loop.

**WITH STRUCTURE**

Multiple properties can be set and multiple methods can be called by using the with...End With statement. The code is executed more quickly and efficiently as the object is evaluated only once.

**Syntax**

With object

[ statements ]

End With

**For Example,**

Here's some code that sets the properties of a Command Button, and then executes its Move Method.

Private Sub Form\_Load()  
Command1.Caption = "OK"  
Command1.Visible = True  
Command1.Top = 200  
Command1.Left = 5000  
Command1.Enabled = True  
Command1.Move 1000,1000  
End Sub

By using the With...End With statement, we can eliminate a few programmer keystrokes, and make the code a bit more readable in the process.

Private Sub Form\_Load()  
 With Command1  
 .Caption = "OK"  
 .Visible = True  
 .Top = 200  
 .Left = 5000  
 .Enabled = True  
 .Move 1000,1000  
 End With  
End Sub

**INTRODUTION TO STANDARD CONTROLS**

In Visual basic 6 standard project the default controls are displayed in the Tool box. It looks like as follows,



**To draw a control using the Toolbox**

1. Click the tool for the control you choose to draw — in this case, the **text box**.

2. Move the pointer onto your form. The pointer becomes a cross hair.

3. Place the cross hair where you want the upper-left corner of the control.

4. Drag the cross hair until the control is the size you want. (*Dragging* means holding theleft mouse button down while you move an object with the mouse.)

5. Release the mouse button. The control appears on the form.



**Command buttons**

* The Command Button is used to initiate actions, usually by clicking on it. The Caption property determines the text to display on the face of the button.
* The Default property, if set to true, means that the button will be activated (same as Clicked) if the <Enter> key is hit anywhere in the form.
* If Cancel is set to True, the button will be activated from anywhere in the form by the <Esc> key.

****

**Properties**

|  |  |
| --- | --- |
| **Properties** | **Description** |
| Appearance | It have the Dimension value(0-3D) |
| BackColor | It sets the background |
| Cancel | [CancelButton](http://msdn.microsoft.com/en-us/library/system.windows.forms.form.cancelbutton(v=vs.90).aspx) |
| Caption | [Text](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.text(v=vs.90).aspx) |
| Font  FontBold  FontItalic  FontName  FontSize  FontStrikethrough  FontUnderline | Font Changes |
| Height | [Height](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.height(v=vs.90).aspx), [Size](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.size(v=vs.90).aspx) |
| Index | Index value for positon. |
| Left | [Left](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.left(v=vs.90).aspx) |
| Top | [Top](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.top(v=vs.90).aspx) |
| Visual Basic 6.0 | Description |
| DragDrop  DragOver | Event when generate when the mouse is click. |
| GotFocus | [Enter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enter(v=vs.90).aspx) |
| LostFocus | [Leave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.leave(v=vs.90).aspx) |

**Text Boxes**

* The text box is the standard control for **accepting input from the user** as well as to display the output.
* It can handle string (text) and numeric data but not images or pictures.
* String in a text box can be converted to a numeric data by using the function Val(text). The following example illustrates a simple program that processes the input from the user.

**Example**

* In this program, two text boxes are inserted into the form together with a few labels.
* The two text boxes are used to accept inputs from the user and one of the labels will be used to display the sum of two numbers that are entered into the two text boxes.
* Besides, a command button is also programmed to calculate the sum of the two numbers using the plus operator.
* The program use creates a variable sum to accept the summation of values from text box 1 and text box 2.
* The procedure to calculate and to display the output on the label is shown below. The output is shown as follows,

Private Sub Command1\_Click()

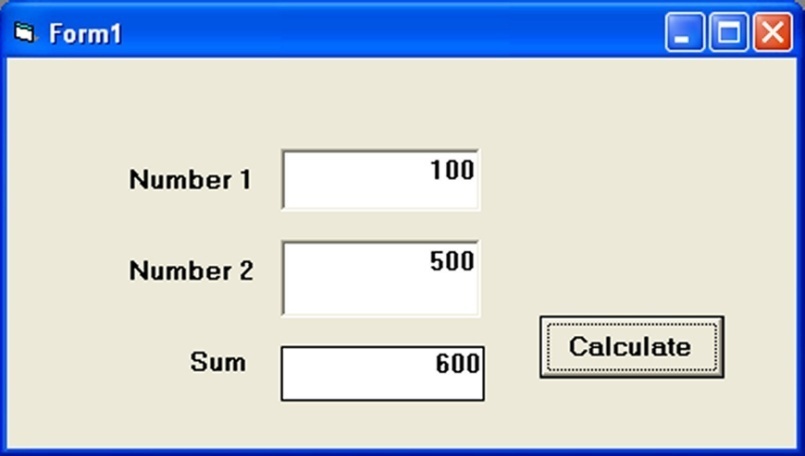
‘To add the values in text box 1 and text box 2

Sum = Val(Text1.Text) + Val(Text2.Text)

‘To display the answer on label 1

Label1.Caption = Sum

End Sub

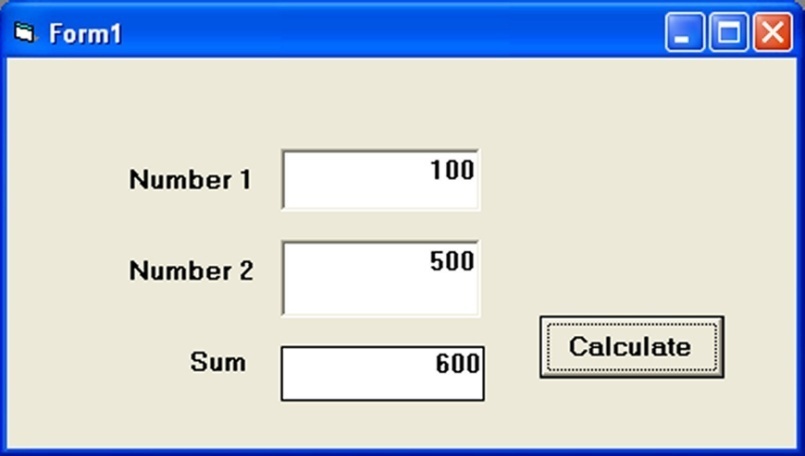


**The Label**

It is used to display static text, titles and screen output from operations. The important properties are,

* Caption - the text that is displayed in the label
* BackColor and ForeColor - colors of the background and the text
* BackStyle - Opaque or Transparent - whether the background is visible or not
* Font - font and size of text
* Alignment - text centered, left or right
* Multiline- True or False - if True, you can have several lines of text, delimited by <CR> in the label - by default, it is set to False

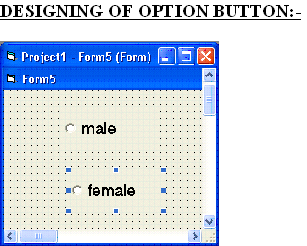
**Example**



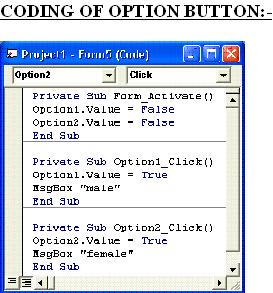
Here the Text **Number 1** and **Number 2** are displayed in the **Label** control only.

**Option Button**

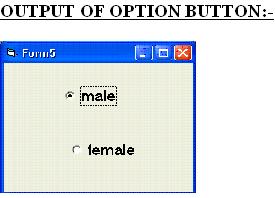
* An Option Button is also known as Radio button.
* An Option button is a round radio button that you select to turn an option on or off.
* This button is used where you have only one option to choose like **yes or no, male or female and married or  unmarried.**



 Double click on **Radio button**  and enter the following code



Press <F5> to run.

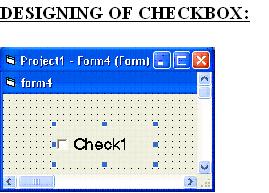


When you click on male button then, it opens the message box.



**Checkbox**

* A checkbox control is rather similar to an option button. The value property of both the controls is tested to check the current state. Check boxes are not mutually exclusive.
* A check box is a control that allows the user to set or change the value of an item as **true or false.** The control appears as a small square.
* When this empty square is clicked, it gets marked by a check symbol . These two states control the checkbox as checked or unchecked.

****

Double click on Checkbox to open code window and type the following code

Private Sub Check1\_Click()

If Check1.Value = 1 Then

MsgBox “red”

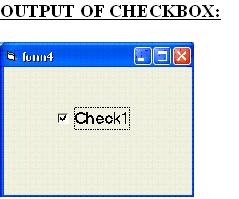
Else

MsgBox “blue”

End If

End Sub

****

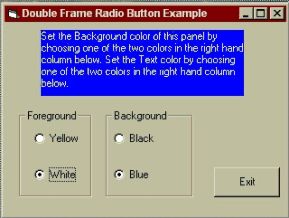
****

When you select the checkbox output is:

****

**Frame Control**

* A **frame** is an object that acts as a container for controls. it is an object that has solid border around it and a caption at the top. When you want to group several controls together - name and address, for example - you use a Frame.
* The frame backcolor can be the same as the form's and only the frame borders will be obvious, or it can be a different color and stand out.
* You create the frame before the controls. When you create controls in a frame, they are tied to the frame and move with it. The frame caption is the text that appears at the top of the frame - you use it to define the group.
* In the Double Frame Radio Button Example below, note the rectangles surrounding both the Foreground and Background option buttons. They're called "Frame Controls".
* They're used for separating the two groupings so that, in each group, the user can select one choice.
* These Frame Controls have properties associated with them just as any other object in Visual Basic. Change the Caption property and you change the title of the Frame Control

****

**List Boxes**

* A ListBox control, displays a list of items. If the total number of items exceeds the number that can be displayed, a scroll bar is automatically added to the ListBox control.
* To add items in a ListBox control, use the Items.Add, for example:
* The following code places "Germany," "India," "France," and "USA" into a list box named List1:

Private Sub Form\_Load ()

List1.AddItem "Germany"

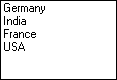
List1.AddItem "India"

List1.AddItem "France"

List1.AddItem "USA"

End Sub

* Whenever the form is loaded at run time, the list appears as shown as follows:

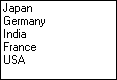


**Adding an Item at a Specified Position**

* To add an item to a list at a specific position, specify an index value for the new item. For example, the next line of code inserts "Japan" into the first position, adjusting the position of the other items downward:

List1.AddItem "Japan", 0

The output is



**Removing Items from a List**

* + For example, to remove the first entry in a list, you would add the following line of code: List1.RemoveItem 0
  + To remove all list entries in bound or standard versions of the list and combo boxes, use the Clear method:

List1.Clear

**Combo Box**

A combo box control combines the features of a text box and a list box. This control allows the user to select an item either by typing text into the combo box, or by selecting it from the list.

**The combo box control**

IC111431

* Combo boxes present a list of choices to the user.
* If the number of items exceeds what can be displayed in the combo box, scroll bars will automatically appear on the control.
* The user can then scroll up and down or left to right through the list.

**When to Use a Combo Box Instead of a List Box**

* Generally, a combo box is appropriate when there is a list of suggested choices, and a list box is appropriate when you want to limit input to what is on the list.
* A combo box contains an edit field
* In addition, combo boxes save space on a form. Because the full list is not displayed until the user clicks the down arrow (except for Style 1, which is always dropped down), a combo box can easily fit in a small space where a list box would not fit.

**Adding Items**

* To add items to a combo box, use the **AddItem** method.

Private Sub Form\_Load ( )

Combo1.AddItem "USA"

Combo1.AddItem "Canada"

Combo1.AddItem "Australia"

  Combo1.AddItem "Brazil"

End Sub

* Whenever the form is loaded at run time, the list appears as shown



**Adding an Item at a Specified Position**

To add an item to a list at a specific position, specify an index value after the new item. For example, the next line of code inserts "India" into the first position, adjusting the position of the other items downward:

Combo1.AddItem "India", 0

**Removing Items**

* You can use the RemoveItem method to delete items from a combo box. RemoveItem has one argument, index, which specifies the item to remove
* For example, to remove the first entry in a list, you would add the following line of code:

Combo1.RemoveItem 0

**To remove all list entries in a combo box, use the Clear method**

Combo1.Clear

**Using Image Control**

The image control is used to display graphics. Image controls can display graphics in the following formats: bitmap, icon, metafile, enhanced metafile, or as JPEG or GIF files.

**The Image control**

IC32185

In addition, image controls respond to the Click event and can be used as a substitute for command buttons, as items in a toolbar, or to create simple animations.

**When to Use an Image Control Instead of a Picture Box Control**

* The image control uses fewer system resources and repaints faster than a picture box control, but it supports only a subset of the picture box control's properties, events, and methods.
* Both controls support the same picture formats. However, you can stretch pictures in an image control to fit the control's size.
* You cannot do this with the picture box control.

**Supported Graphic Formats**

The image control can display picture files in any of the following standard formats like,

Bitmap, Icon, Cursor, Metafile, JPEG, GIF

**Loading a Graphic Into the Image Control**

* Pictures can be loaded into the image control at design time by selecting the Picture property from the control's Properties window, or at run time by using the Picture property and the LoadPicture function.
* Set Image1.Picture = LoadPicture("c:\Windows\Winlogo.cur", vbLPLarge, vbLPColor)
* When a picture is loaded into the image control, the control automatically resizes to fit the picture regardless of how small or large the image control was drawn on the form.
* You may want to use icon (.ico) and cursor (.cur) files containing separate images at different sizes and color depths to support a range of display devices. The LoadPicture function's settings allow you to select images with specific color depths and sizes from an .ico or .cur file. In cases where an exact match to the requested settings isn't available, LoadPicture loads the image with the closest match available.
* To clear the graphic from the image control, use the LoadPicture function without specifying a file name. For example:

Set Image1.Picture = LoadPicture

* This will clear the image control even if a graphic was loaded into the Picture property at design time.

**Picture Box**

The picture box control is used to display graphics, to act as a container for other controls, and to display output from graphics methods or text using the Print method.

**The picture box control**

IC144163

* The picture box control is similar to the image control in that each can be used to display graphics in your application — each supports the same graphic formats.
* The picture box control, however, contains functionality which the image control does not, for example: the ability to act as a container for other controls and support for graphics methods.

**Supported Graphic Formats**

The picture box control can display picture files in any of the following formats: bitmap, cursor, icon, metafile, enhanced metafile, or as JPEG or GIF files.

**Loading a Graphic Into the Picture Box Control**

* Pictures can be loaded into the picture box control at design time by selecting the Picture property from the control's Properties window, or at run time by using the Picture property and the LoadPicture function.
* Set Picture1.Picture = LoadPicture("c:\Windows\Winlogo.cur", vbLPLarge, vbLPColor)

**Graphics Methods**

* Picture boxes, like forms, can be used to receive the output of graphics methods such as Circle, Line, and Point.
* For example, you can use the Circle method to draw a circle in a picture box by setting the control's AutoRedraw property to True.

Picture1.AutoRedraw = True

Picture1.Circle (1200, 1000), 750

**Timer Control**

* Timer controls respond to the passage of time. They are independent of the user, and you can program them to take actions at regular intervals.
* A typical response is checking the system clock to see if it is time to perform some task.
* Timers also are useful for other kinds of background processing.

**The timer control**

Timer

* Each timer control has an Interval property that specifies the number of milliseconds that pass between one timer event to the next.
* Unless it is disabled, a timer continues to receive an event (appropriately named the Timer event) at roughly equal intervals of time.

**Placing a Timer Control on a Form**

* Placing a timer control on a form is like drawing any other control:
* Click the timer button in the toolbox and drag it onto a form.
* The timer appears on the form at design time only so you can select it, view its properties, and write an event procedure for it.
* At run time, a timer is **invisible** and its position and size are irrelevant.

**Initializing a Timer Control**

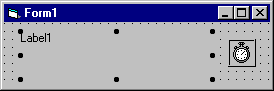
A timer control has two key properties.

|  |  |
| --- | --- |
| **Property** | **Setting** |
| Enabled | If you want the timer to start working as soon as the form loads, set it to True. Otherwise, leave this property set to False. You might choose to have an outside event (such as a click of a command button) start operation of the timer. |
| Interval | Number of milliseconds between timer events. |

Note that the Enabled property for the timer is different from the Enabled property for other objects. With most objects, the Enabled property determines whether the object can respond to an event caused by the user. With the Timer control, setting Enabled to False suspends timer operation.

**Sample Code**

The following code displays the digital time.

****

Private Sub Timer1\_Timer ()

If lblTime.Caption<>CStr(Time) Then

lblTime.Caption = Time

End If

End Sub

**Scroll bars**

* Scroll bars provide easy navigation through a long list of items or a large amount of information by scrolling either horizontally or vertically within an application or control.
* Two types of scroll bars are there in the Visual Basic environment -- HScrollBar (Horizontal Scroll Bar) and VScrollbar(Vertical Scroll Bar).

**The horizontal and vertical scroll bar controls**

IC111803IC52423

The horizontal and vertical scroll bar controls are not the same as the built-in scroll bars found in Windows or those that are attached to text boxes, list boxes, combo boxes, or MDI forms within Visual Basic. Those scroll bars appear automatically whenever the given application or control contains more information than can be displayed in the current window size (or, in the case of text boxes and MDI forms, when the ScrollBars property is also set to True).

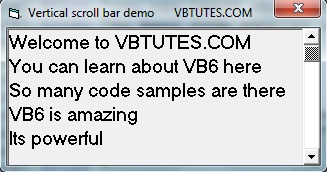
**How the Scroll Bar Controls Work**

The scroll bar controls use the Scroll and Change events to monitor the movement of the scroll box (sometimes referred to as the thumb) along the scroll bar.

|  |  |
| --- | --- |
| **Event** | **Description** |
| Change | Occurs after the scroll box is moved. |
| Scroll | Occurs as the scroll box is moved. Does not occur if the scroll arrows or scroll bar is clicked. |

Using the Scroll event provides access to the scroll bar value as it is being dragged. The Change event occurs after the scroll box is released or when the scroll bar or scroll arrows are clicked.

**Sample program**



**Code**

Private Sub Form\_Load()

Form1.Show

Picture1.Print "Welcome to VBTUTES.COM"

Picture1.Print "You can learn about VB6 here"

Picture1.Print "So many code samples are there"

Picture1.Print "VB6 is amazing"

Picture1.Print "Its powerful"

Picture1.Print "\_\_\_\_\_\_\_\_\_\_\_\_"

Picture1.Print "VB6 is interesting"

Picture1.Print "Visual Basic is awesome"

Picture1.Print "Visit WWW.VBTUTES.COM"

Picture1.Print "...................."

Picture1.Print "...................."

Picture1.Print "----------------------"

End Sub

Private Sub VScroll1\_Change()

Picture1.Move 0, -VScroll1.Value \* 50

End Sub

For example, the TextBox control has a scroll bar with it, and it becomes enabled when you set the Multi-Line property of the TextBox control to True.

**File System Controls**

Three of the controls on the ToolBox let you access the computer’s file system. They are DriveListBox, DirListBox and FileListBoxcontrols .

**The Drive ListBox**

* It is used to display a list of drives available in your computer.When you  place this control into the form and run the program,you will be able to select different drive from your computer.
* The drive list box is a drop-down list box. By default, the current drive is displayed on the user's system. When this control has the focus, the user can type in any valid drive designation or click the arrow at the right of the drive list box.
* When the user clicks the arrow, the list box drops down to list all valid drives. If the user selects a new drive from the list, that drive appears at the top of the list box.

**Example: Setting the Drive**

Private Sub Form\_Load()  
    Drive1.Drive = "G"  
 End Sub



**DirListBox Control**

A drop down list box that displays a hierarchical list of directories in the current drive

**File ListBox**

It displays all files in the current directory or folder.

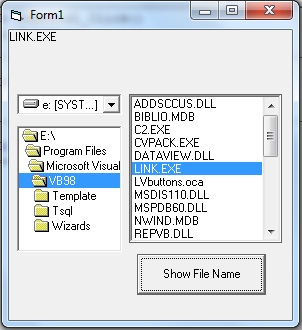
**Working with Drive,DirList,File List boxes**

When the user selects a drive, this change should be reflected in DirListBox and when the user selects a directory, this change should be reflected in the FileListBox. Write the following code for that.   
**Example :**

Program that will display the drives,directories and the files

**Phase I**

Open a form and create the file system controls.The form look like as follows:



**Setting the relationship**

Private Sub Dir1\_Change()

File1.Path = Dir1.Path  
End Sub

* This establishes a connection between the DriveListBox and the DirListBox. The directories for the selected drive are displayed.

Private Sub Drive1\_Change()  
  Dir1.Path = Drive1.Drive  
End Sub

* This establishes a connection between the DirListBox and the FileListBox. All the files in the selected directory are displayed.

**Phase II**

* Adding search criteria,add a textbox control to the form,and add the following code

Private Sub Text1\_Change()  
 File1.Pattern = Text1.Text  
End Sub

* This statement changes the ‘Pattern’ property of the FileListBox, depending upon criteria entered by the user. If the user enters “\*.Doc” in the TextBox only files with “Doc” extension will be displayed.

**UNIT III**

**Introduction to Built-in ActiveX control**

**ActiveX control**

* An ActiveX control is a [component](http://whatis.techtarget.com/definition/component) program  that can be re-used by many application programs
* ActiveX control file has the extension of .OCX (Object Linking and Embedding custom controls).
* It can be developed by [C](http://www.webopedia.com/TERM/C/C.html), [C++](http://www.webopedia.com/TERM/C/C_plus_plus.html), Visual Basic, and [Java](http://www.webopedia.com/TERM/J/Java.html).

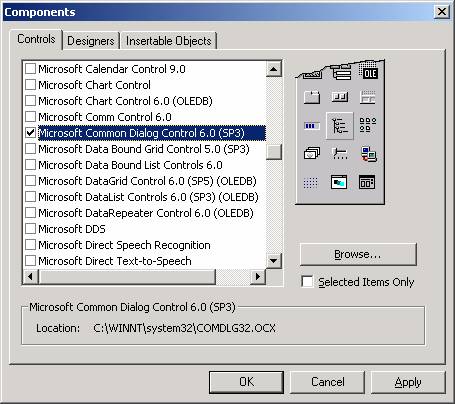
**ActiveX Control List**

The following are the advanced controls:

|  |  |
| --- | --- |
| **S.NO** | **CONTROL NAME** |
| 1 | Toolbar Control |
| 2 | TreeViewControl |
| 3 | ListView Control |
| 4 | ImageList Control |
| 5 | Common Dialog |
| 6 | StatusBar Control |
| 7 | RichText Control |
| 8 | Slider Control |
| 9 | Grid Control |

**ActiveX control in Tool box**

* By default, ActiveX Control doesn’t add in Toolbox
* To place it in tool box, right-mouse click on the toolbox and go to "Components", select the controls and press "OK"



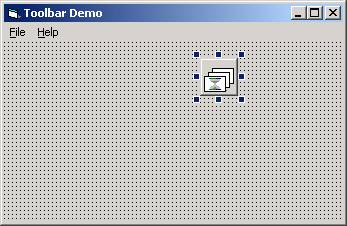
**Components Dialog Box**

**The Image list control**

* An image list control contains a collection of images that is used by other controls like ListView, TreeView and Toolbar controls
* To use with other controls, first a picture can be assigned using the picture property
* The control uses bitmap (.bmp), cursor (.cur), icon (.ico), JPEG (.jpg) or GIF (.gif) files in the collection of ListImage objects
* ListImage object has the properties of **Key** and **Index**
* It has methods, such as **Add**, **Remove** and **Clear**
* This control is not directly access by the user and is not visible at run-time

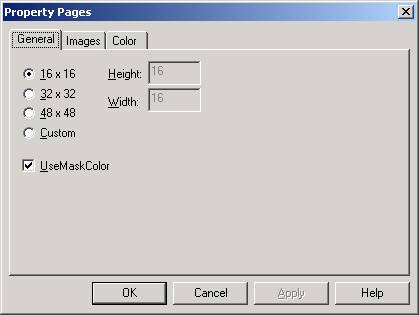
**Adding images to the ImageList**

* Start the new project
* Add the ImageList control by Double-click on the control in the toolbox



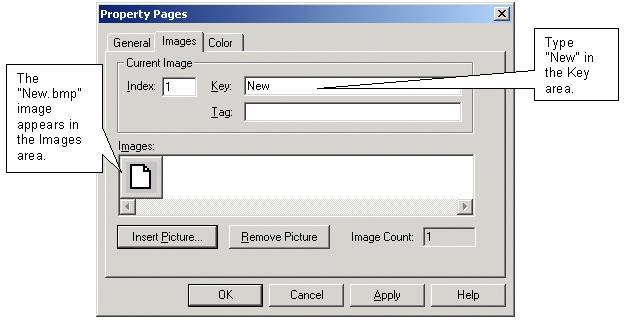
**To add ListImage objects at design time**

* Right-click the ImageList control and click Properties or Press F4 to bring up the Property pages, such as



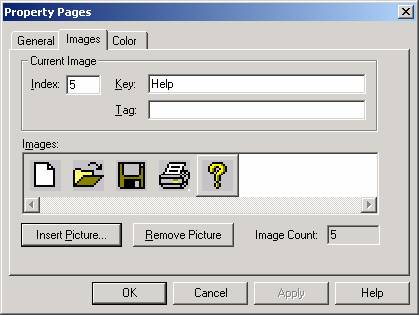
**The General tab of the Property Pages dialog box for the ImageList control.**

* Set the height and width of the images in General tab
* Click the Images tab to add the image details
* Click Insert Picture button to display the **select picture** dialog box
* Select the image and click open
* Enter the **Key** and **Index** value of the selected image that should be a different one
* The Index of each image is assigned by Visual Basic. The Index starts at 1 for the first image and increments by 1 for each additional image



**The Images tab of the Property Pages dialog box for the ImageList control.**

* Add the images for Open, Save, Print, Help
* At the last step, the Property Pages dialog box look like this:



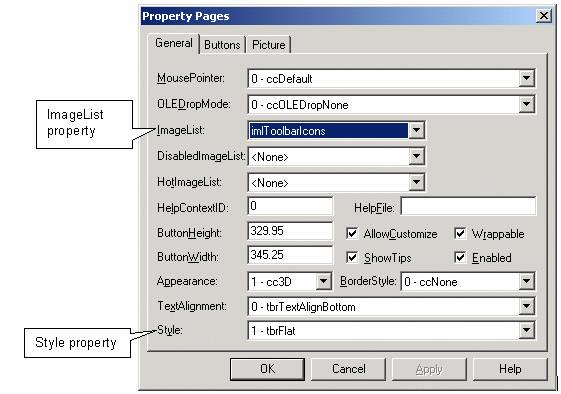
* Click OK to **save and close** the Property Pages dialog box

**ToolBar Control**

* Toolbar control is used to create a toolbar such as open, save, save as etc
* This control has the link with the ImageList control because it uses the imagelist control’s images
* Double-click on the ImageList control to place it onto the form
* Press F4 to display the property page
* To add the images in ImageList , Click the Images tab
* Click insert picture to insert the pictures with the index, key value to identify the individual images

**Setting the Toolbar control property**

* Select the toolbar and press F4 or right-click the toolbar and select Properties to bring up the Property Pages dialog
* On the General tab, set ImageList to "imlToolbarIcons". The toolbar get its images from the ImageList control



**The General tab of the Property Pages for the Toolbar control.**

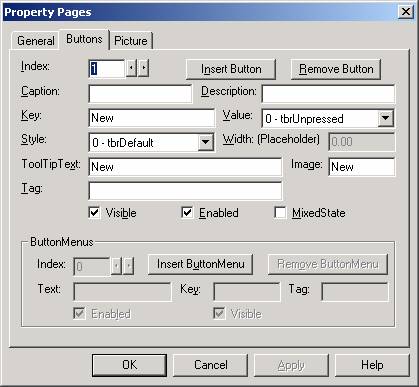
* Set the Styleproperty to "1 – tbrFlat or "0 – tbrStandard"
* The difference between the Standard and Flat Toolbar Style is as follows,

Standard Flat

image014 (1) image015 (1)

* Click the Buttons tab of the Property Pages dialog box, it look like this:

**The Buttons tab of the Property Pages for the Toolbar control.**



* Click the Insert Button  and set the following properties:

**i. Key**

Set the Key property to "New".

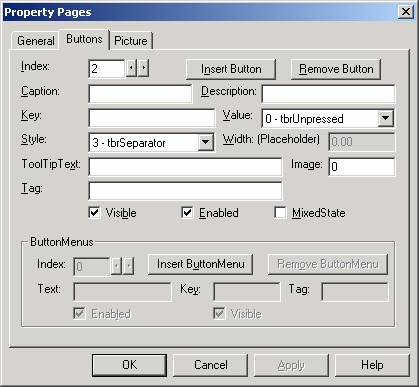
**ii. ToolTipText**

Set the ToolTipText property to "New". This is text that will pop up in a little yellow label when the user moves the mouse over the button

**iii. Image**

Set the Image property to "1" or "New". This value is equal to ImageList control’s image property

* For Flat toolbar Style, it has vertical line separates each button in the toolbar. To add this, click the "Insert Button" button again
* Leave the values for Key, ToolTipText, and Image properties, but set the Styleproperty to "3 – tbrSeparator"
* The Property Pages dialog box should look like this:

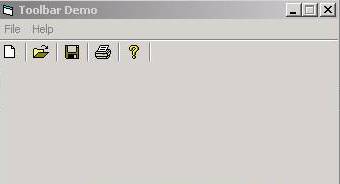


**The Buttons tab of the Property Pages for the Toolbar control.**

* This process will continue until all icons are added
* **Example**: Set the values for Index, Key, Style, ToolTipText, Images propery in this order:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **Key** | **Style** | **ToolTipText** | **Image** |
| 1 | New | 0 – tbrDefault | New | 1 (or "New") |
| 2 |  | 3 - tbrSeparator |  |  |
| 3 | Open | 0 – tbrDefault | Open | 2 (or "Open") |
| 4 |  | 3 - tbrSeparator |  |  |
| 5 | Save | 0 – tbrDefault | Save | 3 (or "Save") |
| 6 |  | 3 - tbrSeparator |  |  |
| 7 | Print | 0 – tbrDefault | Print | 4 (or "Print") |
| 8 |  | 3 - tbrSeparator |  |  |
| 9 | Help | 0 – tbrDefault | Help | 5 (or "Help") |
| 10 |  | 3 - tbrSeparator |  |  |

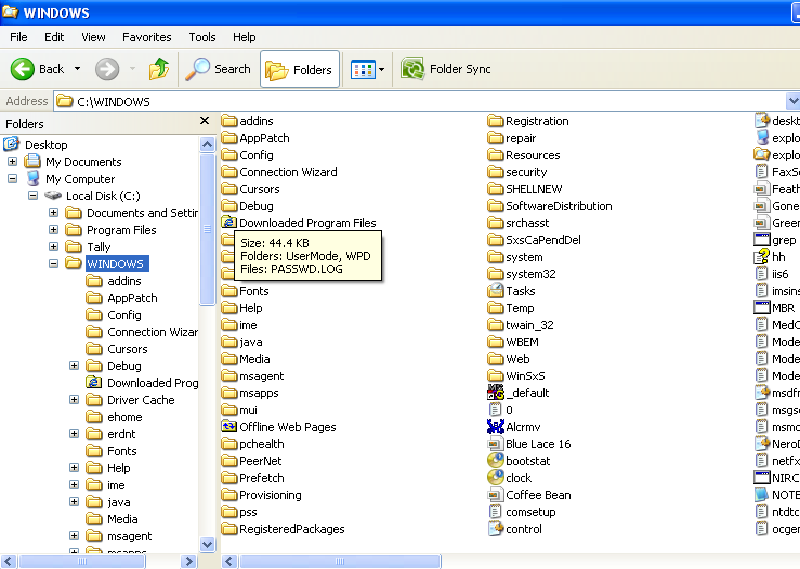
* Click OK to save and close the Property Pages dialog box
* Run the program. It displays the output as follows



**Tree View Control**

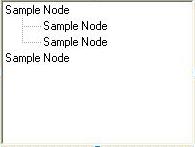
* This control is used to display the data in hierarchical view (left side panel)
* Windows explorer is built using TreeView control
* It displays the drives, directories, subdirectories and files in the form of hierarchy.
* It uses the ImageList control’s images

**Example:**



**Creating a TreeView Control**

* Choose the  “TreeView” control from the tool box and place it in the form. It looks like as follows



* Each item on the Tree View is a Node
* A Node can have Child Nodes
* A node can be a child node, a parent node or a sibling to another node
* To add the node use the add method

**Syntax**

object.Add (relative,relationship,key,text,image,selectedimage)

**Parameters**

object : The name of the TreeView control

relative : The index number or key of a previous node

relationship : Optional. The relationship of the new node with the previous node

key : A string that is used to retrieve the Node

text : The name of the node

image : Optional. The index of an image in imagelist control

**Example**

tv1.Nodes.Add “state”, tvwChild, “City”

**Methods**

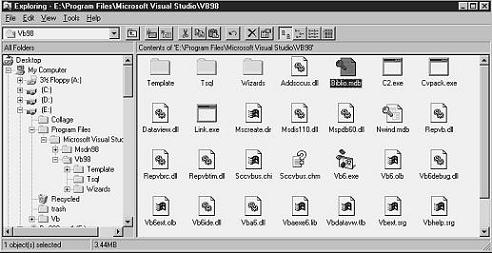
* tvwLast – The Node is placed after all other nodes at the same level of the node named in relative
* tvwNext – The Node is placed after the node named in relative
* tvwPrevious - The Node is placed before the node named in relative
* tvwChild – The Node becomes a child node of the node named in relative

**Example:**

|  |
| --- |
| tvwCustTree.Nodes.Add “State”, tvwNext, “country”  **Tree definitions** |
| |  |  | | --- | --- | | **Node** | A node can be best thought of as a branch on our tree. Each node represents 1 item. A node can hold any number of subnodes. | | **Sibling** | A sibling is another node that exists on the same branch as the referring node. | | **Children** | Children are sub-nodes of the current node. Sometimes called Child nodes. | | **Parent** | A parent node is the node "up" from your current node. All nodes except the root node have a parent node. | |

**The List view control**

* The ListView control displays lists of information to a user
* Windows Explorer provides an example of the ListViewcontrol
* The right side contains a list of items(Listview) within a directory



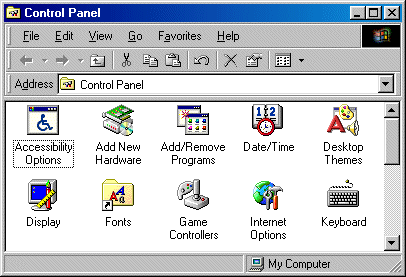
**Example of a ListView in Windows Explorer showing large icons**

* Items can be displayed in any of 4 types
* **Types**
* **LargeIcon** - Displays items with large icons and the main text
* **Small Icon** - Displays items with small icons and the main text
* **List** - Displays items with the main text
* **Details** - Displays items with small icons, the main text, and any other data to be displayed in columns

**LargeIcon**

In this view, each item is displayed with 32x32 pixels icon. The string of the item displays under its corresponding icon:

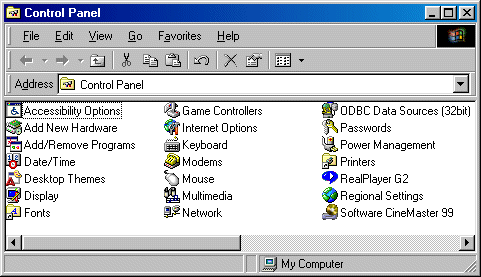
**Example:**



**List**

Each item appears with the 16x16 pixels small icon to its left

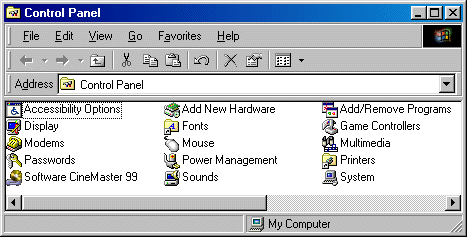
**Example:**



**Small Icon**

Each item appears with the 16x16 pixels icon

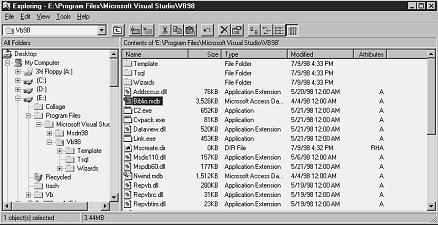
**Example:**



**Detail**

Each item and its detailed data appears in a single row

**Example:**



**The Common Dialog box Control**

* This control allows to create the dialog boxes
* It allows to display the following dialog boxes,
  + Open a file
  + Save a file
  + Set a color
  + Set a font
  + Print a Document
* Add the control in the form and call the methods
* ShowOpen
* ShowSave
* ShowColor
* ShowFont
* ShowPrinter
* ShowHelp

**Working with the Common Dialog Control**

* Add the common dialog control to the toolbox by using Components dialog box
* Draw it in the form
* During run time it is not visible, because this method can be invoked during runtime.

**The File Open Dialog Box**

**Example**

It provides simple code to allow the user to open a file

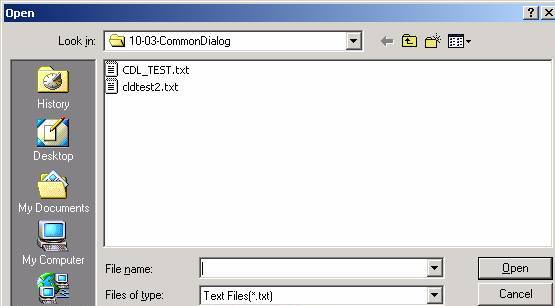
**Syntax**

CommonDialog1.ShowOpen

**ShowOpen**

* The ShowOpen method is used to display the dialog to open files
* The FileName Property is used to determine the name of the file the user selected
* The Filter Property is used to determine the types of file to be shown. A Pipe symbol, |, is used to separate the description and value

CommonDialog1.Filter = "Text (\*.txt)|\*.txt"



**Open file Dialog box**

**The File Save Dialog Box**

**Example**

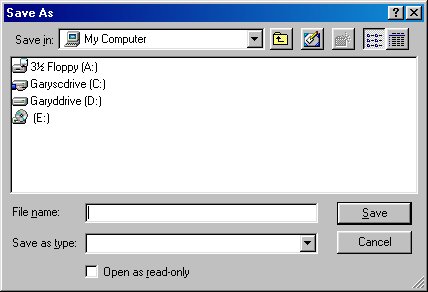
It provides asimple code to allow the user to save a file

**Syntax**

CommonDialog1. ShowSave

**ShowSave**

The ShowSave method looks exactly the same as the ShowOpen method, except the default title is "Save As" rather than "Open"



**Save file Dialog box**

**The File Color Dialog Box**

**Example:**

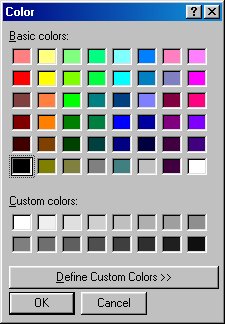
It gives a simple code to allow the user to display color dialog box

**Syntax**

CommonDialog1. ShowColor

**ShowColor**

|  |  |
| --- | --- |
| * The ShowColor method is used to display the Color dialog box * The **Color** property is used to determine which color was selected * **Example**  |  | | --- | | CommonDialog1.ShowColor  Picture1.FillColor = CommonDialog1.Color | |
|  |



**ShowColor Dialog box**

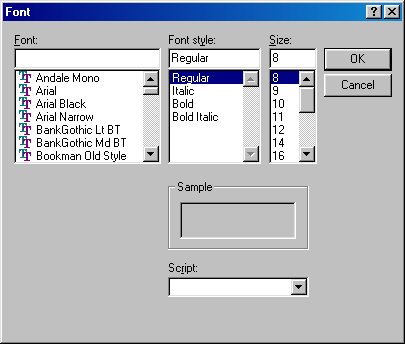
**The File Font Dialog Box**

**ShowFont**

* The ShowFont method is used to display a list of fonts that are available
* Before the ShowFont method can be displayed, the Flags property must be set to one of the following names or a Message Box is displayed with a message of "There are no fonts installed", and a run-time error occurs
* cdlCFScreenFonts
* cdlCFPrinterFonts
* cdlCFBoth
* **Example**

CommonDialog1.Flags= cdlCFScreenFonts

CommonDialog1.ShowFont



**ShowFontDialog box**

**The File Help Dialog Box**

**Showhelp**

* The ShowHelp method is used to display a Windows Help File
* The HelpFile and HelpCommand properties must be set in order for the Help file to be displayed  
   CommonDialog1.HelpFile = "c:\juicystudio\tutorial.hlp"  
   CommonDialog1.HelpCommand = cdlHelpContents  
   CommonDialog1.ShowHelp

**The File Printer Dialog Box**

|  |
| --- |
|  |
| * **ShowPrinter**  The ShowPrinter method is used to display the dialog to open files * The following example uses the ShowPrinter method with the Printer object to set the number of copies and page orientation   CommonDialog1.ShowPrinter  Printer.Copies = CommonDialog1.Copies  Printer.Orientation = CommonDialog1.Orientation  Printer.EndDoc  This control helps to assign the properties for Printer also |
|  |

**Status bar control**

* The status Bar control contains a collection of Panel objects, each of which can display text and images
* The status bar contains maximum of sixteen panels or frames.
* Each of these panels gives different types of information like the time, Caps Lock,Scroll Lock,NUM lock, purpose of the control or Text Type or Maximum number of Characters etc.
* The status bar control can be placed at the top, bottom and sides of an application

**Working with Status bar control**

* Draw the status bar on your form. It will automatically get aligned to the bottom of the form. By default it will show a single panel

**The Panel Object and the Panel Collection**

* The status bar control is built around the Panel collection. It can add or remove each panel at design time or run time
* To add panel **at design time,** right-click on the control and click on properties to display the properties pages dialog box
* **At run time**, dynamically change the text, images or widths of any panel object, using the text, picture and width properties

**Creating a Panels in StatusBar control**

* The status bar control is named “StatusBar1”

Dim Pan1 as Panel

Set Pan1 = StatusBar1.Panels.Add()

Add method is used to add the Panel in a StatusBar control

**Example:**

Private Sub Text1\_GotFocus()

StatusBar1.Panels.Item(0).Text=”Enter the Item Number with numeric value”

End Sub

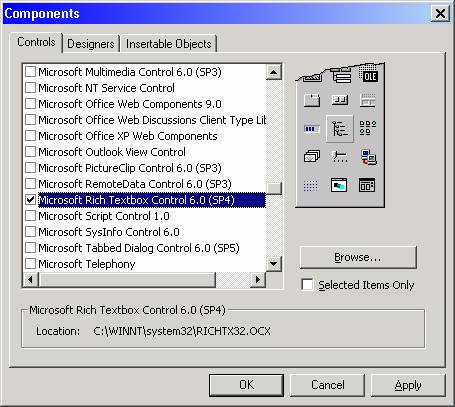
It displays the help or hints for the Item Number Text box’s purpose and the text type.

**Rich Textbox Control**

* The RichTextBox control allows the user to enter and edit text with more advanced formatting features than the TexBox control
* The Textbox can hold only plain text
* The text can be change like, text bold, italic, change the color and create superscripts and subscripts, insert tables, insert pictures, Highlight text, add a animated image etc

**Working with Rich Textbox Control**

* To place it in tool box, right-mouse click on the toolbox and go to "Components", select the Microsoft Rich Textbox control
* Press OK to save and close the components box



* Rich TextBox control added in a Toolbox with the icon like image002

**Properties of the RichTextBox Control**

* Locked
* MaxLength
* MultiLine
* ScrollBars
* SelBold
* SelItalic
* SelUnderline
* SelStrikethrough

**Methods of the RichTextBox Control**

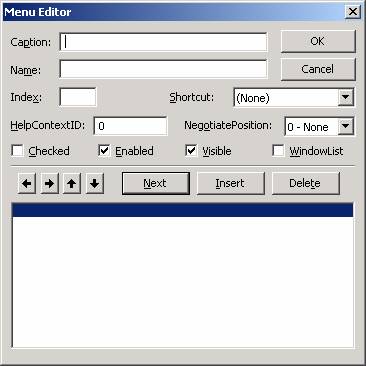
* LoadFile
* SaveFile
* SelPrint

**Event**

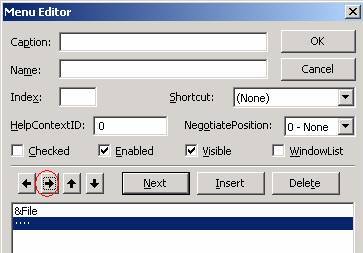
* SelChange Event

**Menu editor**

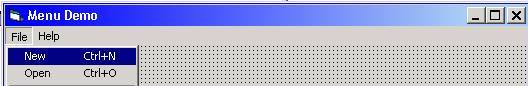
* It is used to create a menu bar
* To place the menu in the forms **choose the menu Editor** option **from the Tool menu**
* The screen appears, as shown below:



* For "Caption", type &File, For "Name", type menu File
* Click the Next button
* Click the "right-arrow" button (shown circled below). A ellipsis (...) will appear as the next item in `the menu list, indicating that this item is a level-two item (below "File")



* For "Caption", type &New; for "Name", type mnuNew, and for "Shortcut", select Ctrl+N. Click the Next button
* For "Caption", type &Open; for "Name", type mnuOpen, and for "Shortcut", select Ctrl+O
* At this point, we are done creating our menu entries, so click the OK button. That will dismiss the menu editor and return focus to the VB IDE
* It looks like as follows:



* Click on the New menu item. The code window for the mnuFileNew\_Click event opens, as shown below:



* Place the code to execute when the user clicks the New menu item
* **Example**:

MsgBox "Code for 'New' goes here.", vbInformation, "Menu Demo"

* During runtime when the click a New menu /Press Ctrl+N , it will open a message box with the message of “Code for 'New' goes here”

**UNIT – IV**

DDE Properties – DDE Methods – OLE properties – ActiveX Control Creation and Usage and ActiveX DLL creation and usage – Database access– Data Control– Data grid( Field control )record set using SQL to manipulate data– Open Data Base Connectivity.

**Dynamic Data Exchange (DDE)**

* Dynamic Data Exchange (DDE) enabled the exchange of information between applications
* It also allowed one application to send commands to the other application
* It was replaced by **O**bject **L**inking and **E**mbedding concepts

**DDE properties, methods, and events:**

|  |  |
| --- | --- |
| **Category** | **Name** |
| Properties | LinkItem  LinkMode  LinkTimeOut  LinkTopic |
| Methods | LinkExecute  LinkPoke  LinkRequest  LinkSend |
| Events | LinkClose  LinkError  LinkExecute  LinkNotify  LinkOpen |

**The DDE properties**

* The Destination Control has a Linking Properties,

**LinkItem**

* Returns or sets the data passed to a destination control in a DDE conversation with another application.
* **Syntax**

*object*.LinkItem [= *string*]

# LinkMode

* Returns or sets the type of link used for a DDEconversation and activates the connection using Control and Form
* Control allows a destination control on a Visual Basic form to initiate a conversation, as specified by the control's **LinkTopic** and **LinkItem** properties
* Form allows a destination application to initiate a conversation with a Visual Basic source form.
* **Syntax**

object.**LinkMode** [= number]

* There are three modes:
* **Automatic** The source sends the data to the destination every time it changes
* **Manual**   
   The destination has to explicitly ask for the data - the source only supplies it on request
* **Notify**   
   The source notifies the destination whenever the data changes, but it only supplies the updated data when the destination asks for it

# LinkTimeout

* Returns or sets the amount of time a control waits for a response to a DDE message.
* Syntax

object.**LinkTimeout** [= number]

Here, the number specifies the numeric value for wait time.

# LinkTopic

* Use LinkTopic with the LinkItem property to specify the complete data link.
* Syntax

object.**LinkTopic** [= value]

## DDE Methods

**LinkPoke**

* Transfers the contents of a Label, PictureBox, or TextBox control to the source application in a DDE conversation.
* Syntax

object.LinkPoke

# LinkRequest:

* Asks the source application in a DDE conversation to update the contents of a Label, PictureBox, or TextBox control.
* Syntax

object**.LinkRequest**

# LinkSend

* Transfers the contents of a PictureBox control to the destination application in a DDE conversation.
* Syntax

object.**LinkSend**

# LinkExecute

* Sends a command string to the source application in a DDE conversation.
* Syntax

object.**LinkExecute** string

**OLE (Object Linking and Embedding)**

* This control allows us to link the program to another program as object
* OLE (Object Linking and Embedding) means to interchange data between applications
* A primary use of the OLE Container control was to embed Word documents or Excel spreadsheets in a form.

**Steps to Create OLE Control**

* Select OLE control in Toolbox and place it in the form.
* Insert Object Dialog is displayed to allow to either embed or link an object into OLE container.
* The available options in insert dialog are Create New, where you select an Object Type and create an object using the appropriate application, or Create from File, where you can create an object by selecting a file from file system.

**To embed a word document into OLE Container control**

* In Insert Object Dialog box select **Create from File** radio button.
* Click on Browse button and select a document file. Click Ok.
* An object is embedded into OLE Container control and a part of document is displayed.  
  Run the project using F5.
* Double click on OLE Container control. This action will invoke MS-Word and run it in OLE Container control.
* Make necessary changes using MS-Word.

**Properties of OLE Control**

* **SizeMode**   
   Determines how image is displayed in OLE control. The valid options are

**Option Value Meaning** VbOLESizeClip 0 The object is displayed in orginal size. If the size of object is larger than the size of control then object is clipped to the size of the control.  
 VbOLESizeStretch 1 The size of the object is resized to the size of OLE container control.  
 VbOLESizeAutoSize 2 OLE container is automatically resized to the size of the object.  
 VbOLESizeZoom 3 The object is resized to fill the OLE container control as much as possible while still maintaining the original proportions (width and height) of the object.

* **UpdateOptions** Specifies how a linked object is updated in Container when the original object is modified. The valid options are,  
   **Option Value Meaning** VbOLEAutomatic 0 Object is automatically updated whenever source data is updated.  
   VbOLEFrozen 1 Updates object whenever data in source application is saved.  
   VbOLEManual 2 Object is updated by invoking Update method of OLE Control.
* **AutoActivate property** Specifies how an object is activated. When object is activated the source application is invoked. The valid options are,  
   **Option Value Meaning** VbOLEActivateManual 0 Use Do Verb method to activate object.  
   VbOLEActivateGetFocus 1 If object supports single click activation, and then source application is activated whenever OLE container control receives focus.  
   VbOLEActivateDoubleclick 2 Double clicking on the object activates source application.  
   VbOLEActivateAuto 3 Activation depends on the type of object.
* **MiscFlags Property**  
   Allows you to specify how data is to be stored and whenever in-place activation is enabled. The valid options are,  
   **Option Value Meaning** VbOLEMiscFlagMemStorage  1 Causes the control to use memory to store the object while it's loaded.  
   VbOLEMiscFlagDisableInPlace  2 Overrides the control's default behavior of allowing in-place activation for objects that support it.

**ActiveX Control Creation and Usage**

**ActiveX Control**

* ActiveX controls are created by using existing controls.
* It can be used by other development tools also.

**Creation of ActiveX Control**

* Start a New project. Select ActiveX Control
* The Title Bar display the caption as “User Control1”
* Save this user control. An ActiveX control is saved in a file with .ctl extension
* There will also be a .vbp file
* Open the project again: Click on file -> select Add Project->Select standard exe this time
* Now, the Project explorer has two Projects
* To the standard .Exe project Form 1, add the new user control
* The next step is to add some controls and try the ActiveX control again
* Close this Project and open the UserControl project
* Create a calander that will accept the month and year and display the days. So place the controls,
* One Combobox for the Month,
* One ComboBox for the Year,
* One MSFlexGrid to display the days,
* Two LabelBox Controls to serve as captions.

**Add some properties to the controls**

* For the ComboBox that is going to display/ accept months, we must add the 12 months in the Listproperty
* For the ComboBox that is going to Display the Years add the years from 1990 to 2005
* For the MSFlexGrid, the number of columns is seven and the number of rows is seven, for the first row, add the headings. Sun ; Mon ; Tue ; …Sat
* After setting all the properties of the Control, Visual Basic 6.0 has an ActiveX control interface wizard
* Click on Addins->Select the ActiveX Control Interface wizard
* It opens with the introduction screen.This displays 2 list boxes.Left side list box contains all properties, events and methods.Select the needed objects alone
* Click Next to Proceed
* Here add the properties which are assigned to a control by clicking New button.Click Next to proceed
* It opens with the default attributes for the events/properties.Click Finish to close the wizard
* Open the Code window of the UserControl and write the codings for adding the values for month, year, date values

**Usage of ActiveX Control**

* If the ActiveX Control registered, it can be reused by various areas.
* For Registration, choose Make..OCX from File menu
* It will register the new control in Windows System Registry
* To add this control in Toolbox, go to Windows/System Directory
* Run the command “regsvr32[controlName].ocx

**ActiveX DLL Creation and Usages**

**ActiveX**

* ActiveX technology is based on the Component Object Model (COM). COM is Microsoft’s standard that defines how software components interact with each other.

**DLL**

* Dynamic-link libraries (DLL) contain reusable functions and data.
* With the use of Dll files, you can store often used functions in a common place, link to it through code, and use its functionality in any of your applications without adding the common procedures to every application.

**Creation of DLL**

* Start VB and select ActiveX Dll for the project type.
* In the properties window for the Class, name it 'CSomeDll'.
* Select Project from the menu, select Project1 Properties from the drop down list, change the project name to 'SomeDll' and Description to ‘SomeDll Creation”
* Now lets add a method (Function) to the class.
* Make a Public function named 'ReverseString' in the class module. It must be Public if you want it to be visible to your client applications.  
   Public Function ReverseString(ByRef SomeString As String) As String  
   ReverseString = StrReverse(SomeString)  
   End Function
* Compile it by selecting File, in the drop-down list select Make SomeDll.dll and choose a safe place to save it.
* Save the project wherever you like and close it.

**Usage of ActiveX DLL**

* Start VB and Select 'Standard Exe' for the project type.
* Add one textbox named Text1 and one Command Button named Command1 to the form.
* Now lets reference our Dll by selecting Project from the menu bar, then select References from the drop-down list.
* Scroll through the list and look for ---> ‘SomeDll Creation' <---
* That is the Project Description we gave our Dll in step 4 of creating the Dll.
* Put a check mark by it and click ok.   
  Now our client app knows about the library we want to link to but we still need to assign the class in the Dll to an object variable.
* Add this code to the general declarations section of your forms code module:   
   Dim X As CSomeDll
* 'CSomeDll' is the name of the class in the DLL that has the method we want to use. We named it this in step 2 of creating the Dll.
* Now add this code to Command1:  
   Private Sub Command1\_Click()  
   Set X = New CSomeDll '<---Assigns an object reference  
   Text1.Text = X.ReverseString(Text1)  
   Set X = Nothing '<--------Release resources to the object  
   End Sub
* Run the program and click the button

**Database access**

* Adatabase is a collection of records that can be manipulated and stored in tables.
* It has a set of rules and tools to manage these records.
* There are different types of databases each with its own format.
* The most commonly known database types are:
* Relational Model
* Network Model
* Hierarchical Model

**Table**

* A table is a basic repository in which data is stored and it is made up of rows and columns.

|  |  |  |  |
| --- | --- | --- | --- |
| Sno | Name | Average | Result |
| 101 | Murugan | 89 | Pass |
| 102 | Suba | 78 | Pass |

**Rows**

* The record that contains the information about items in the table.

|  |  |  |  |
| --- | --- | --- | --- |
| 101 | Murugan | 89 | Pass |

**Column**

* The small piece of information about the record is given in column. That is each field in a table is called as column.

|  |
| --- |
| Result |
| Pass |
| Pass |

**Primary Key**

* A primary key is a unique field or a combination of more than one filed that identifies a record.

The duplicate key was not found.

**Index**

* An index is a list that contains the key field of a record and its physical location of a database.

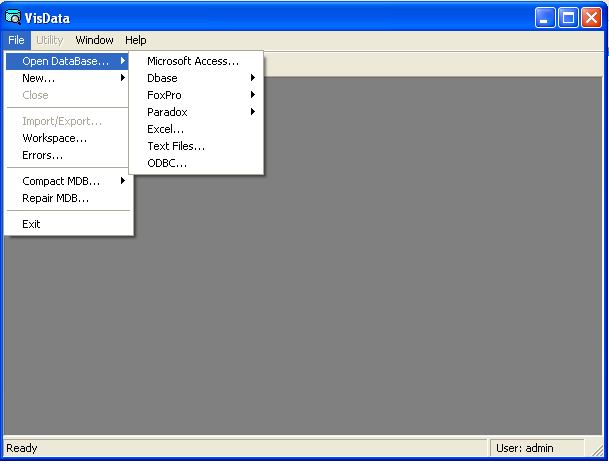
**Creating Table**

There are three ways to create a table.

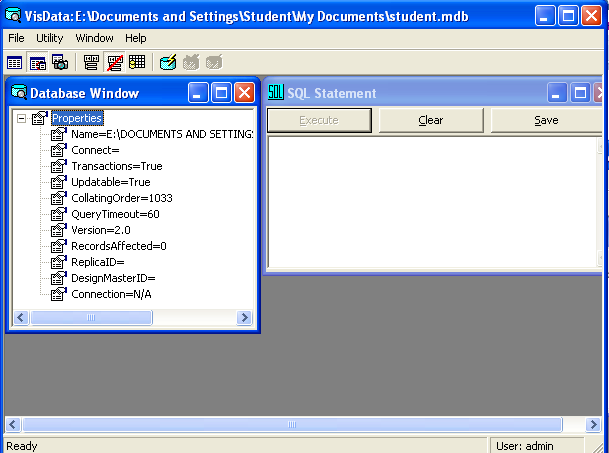
* Using DAO(Data Access Object)
* Using Microsoft Access
* Using Visual Data Manager in VB.

**Working With Visual Data Manager**

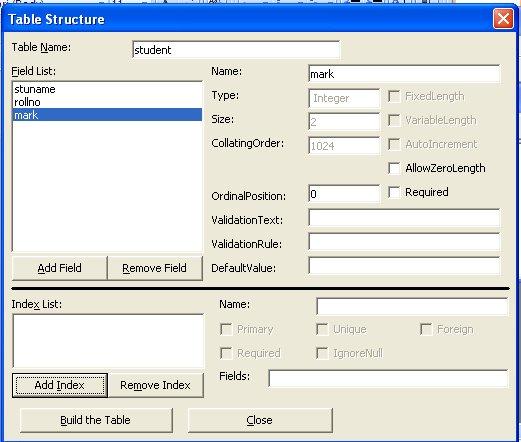
* The visual Data Manager option is available under the Add Ins Menu
* Click on the Visual Data Manager
* The menu items available are File, Utility, Window and Help.
* From the File menu select the option New



* From the pop-up menu select Access and depending on the version of VB
* That is installed in the computer choose the version for MDB
* VB will present a save database dialog box to the database
* VB display two window after the database is created
* Database Window
* SQL Window.
* Database window will list out of the item properties that are available for the database.



* Right click the database window ->select New Tables.
* It appears Table structure window.



* Specify the table name and click on add field button. To create a field name in our table.

|  |  |  |  |
| --- | --- | --- | --- |
| Student | | | |
| Rollno | stuname | Mark | Result |
|  |  |  |  |

* After the table as be created it is required to fill the data in a table.
* Visual data manger displays the database window with the list of table that has been created.
* Double click on the table we want to add the data to the table.
* A small window opens to enter the data.
* It is possible to modify the field name using table structure dialog box.

**Data** **Control**

* Data Control is used to manipulate the database.
* It establishes a connection to a database
* It returns a set of records from the database
* It enables the users to move from record to record
* It also enables the users to display and manipulate data from the records in controls

**Using the Data Control**

* It is available on the Toolbox. It looks like http://graphicsmagician.com/vbcourse/controls/data.gif
* Draw the data control on the form. It appears like



* It has buttons for moving from record to record move to the next record or last record and to move to the previous and first record.

**Setting the Properties for the Data Control**

* Click on the data control and bring up the properties window.
* **Name**

The name of the control. Let us call it Customerdata. Refer the data control with this name.

* **BOFAction**

Set to MoveFirst. When the Beginning of File is reached, the data control must be told to point to the first record.

* **Caption**

Set it to Customers. This will be the title of the data control. The default is Data1.

* **Connect**

Select Access. Access is the default.

* **Database Name**

It specifies the name of the database

* **EOFAction**

Set it to Move Last. This tells the data control to point to the last record when the user reaches the End of the File.

* **Recordset Type**

Select 0-Table type. A Recordset Type Property determines the type of object that will be created by the Data control. The options are

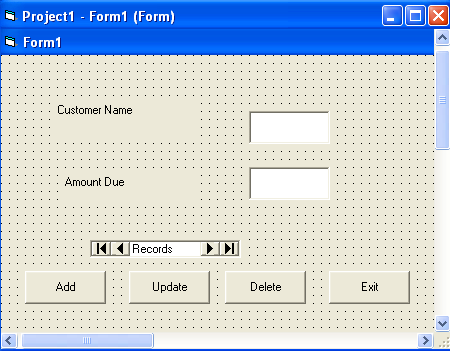
* **Editable Recordset** is created based on data from a single table.
* **Dynaset**, where an editable Recordset is created based on data from one or more tables.
* **Snapshot**, where a non-editable Recordset is created based on data from one or more table
* **Recordsource**

Set it to customer\_Data. In this Property we tell the data control the table or Query that must be used while creating the Recordset.

**Steps to add the value in controls Using DataControl**

* To use a Data Control, first select the control from the toolbox and draw it on the form. Its default name is Data1
* In the properties window, set the DatabaseName property to the filename of the database
* Click on the ellipsis button (...) to use a "file open" dialog
* Set the RecordSource property to the name of a database table within the database selected
* Put a Text Box on the form.
* Set the DataSource property for the text box to Data1
* Set the DataField property for the text box to the field to display.
* If steps 2-5 have been accomplished with a valid database, a drop-down list of choices will be provided.
* Put a Label control in front of the text box with the fieldname selected in step 6
* Repeat steps 4-7 for each field you wish to display
* Run the application

**Example**: To display the Customer name and amount from the database using Data Control



**Adding a New Record**

The AddNew method adds a new record at the end of the file.

Private Sub CmdAdd\_Click ()

Customerdata. Recordset. AddNew

End Sub

**Delete a Record**

The delete method deletes the current record.

Private Sub CmdDelete\_Click ()

Customerdata. Recordset. Delete

End Sub

**Updating a Record**

The datacontrol updating the database with the changes made to the current record.

Private Sub CmdUpdate\_Click ()

Customerdata. UpdateRecord

End Sub

**DataGrid( Field control ) Control**

* The DataGrid control is a spreadsheet-like bound control that displays a series of rows and columns representing records and fields from a Recordset object.
* When set the DataGrid control's DataSource property at design time, the control is automatically filled and its column headers are automatically set from the data source's recordset.
* Then edit the grid's columns; delete, rearrange, add column headers to, or adjust any column's width.

**To implement a DataGrid control at design-time**

* Create a Microsoft Data Link (.UDL) file for the database
* Place an ADO Data control on a form, and set the ConnectionString property to the OLE DB data source created in step 1.
* In the RecordSource field of the Ado Data control, type a SQL statement that returns a recordset.
* For example,

Select \* From MyTableName Where CustID = 12

* Place a DataGrid control on a form, and set the DataSource property to the ADO Data control.
* Right-click the DataGrid control and then click Retrieve Fields.
* Right-click the DataGrid control and then click Edit.
* Resize, delete, or add columns to the grid.
* Right-click the DataGrid control and then click Properties.
* Using the Property Pages dialog box, set the appropriate properties of the control to configure the grid as you wish it to appear and behave.

## Changing Displayed Data at Run Time

* It also allows to dynamically changing the data source of the grid.

Dim strQuery As String

strQuery = "SELECT \* FROM Suppliers WHERE SupplierID = 12"

Adodc1.RecordSource = strQuery

Adodc1.Refresh

### Changing the DataSource

* At run-time DataSource property can be reset to a different data source.
* Set DataGrid1.DataSource = adoPubsAuthors
* Returning Values from the DataGrid
* Once the DataGrid is connected to a database, you may want to monitor which cell the user has clicked.
* Use the RowColChange

Private Sub DataGrid1\_RowColChange (LastRow As Variant, ByVal LastCol As Integer)

‘Print the Text, row, and column of the cell the user clicked.

Debug.Print DataGrid1.Text; DataGrid1.Row; DataGrid1.Col

End Sub

## Structured Query Language

* SQL stands for Structured Query Language
* It is the closest thing to a standard for performing database queries
* Simple SQL Queries
* An SQL query statement looks like this:

SELECT fields FROM table WHERE condition ORDER BY field

* Fields is a list of fields from which data is retrieved. Often \* is used to ask for all fields.
* If individual fields are requested, separate the field names with commas.
* Table is the data table name
* Condition can use relational operators such as =, <, >, or LIKE. Examples:

WHERE Name = "John" Exactly John

WHERE Name LIKE " John \*" John, followed by any other characters

* Sort field is a field on which the sort is performed. Use DESC to sort in descending order. Examples:

ORDER BY Year ascending sort by Year

ORDER BY Rating DESC descending sort by Rating

ORDER BY Year, Rating primary and secondary sort fields

**ODBC(Open Database Connectivity)**

* ODBC is Microsoft interface for accessing data in a heterogeneous environment of relational and non relational database management system.
* With ODBC application developers can allow and application can currently access, view, modify data from multiple databases.
* To use ODBC the following three components are required,
* ODBC client
* ODBC server
* OBDC drivers

**ODBC Client**

* An ODBC enabled front end or client side desktop.
* Application also known as ODBC client and thus see on the computer screen example VB.

**ODBC Server**

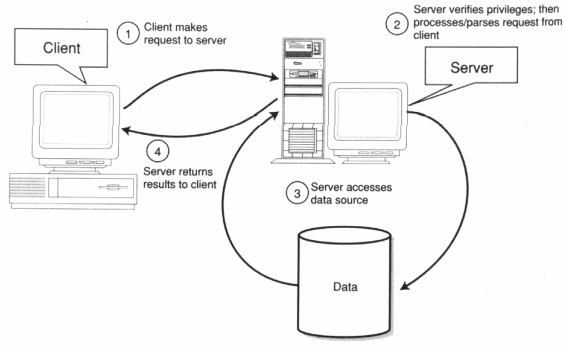
* It is a back end application are server DBMS that is used to retrieve the information from the database and send to ODBC driver.
* Ex: MS-Access.

**ODBC Drivers**

* It is software that resides on the front and it is used to translate the command into a format that ODBC server can understand.
* ODBC server send the answer back to the ODBC driver which translate the answer into a format that client can understand.

**Advantages of ODBC**

* Well integrated in to many RAD tools.
* Interfaces easily into a variety of “data bound” components in development.
* Simplifies and speeds application development.

**ODBC PROCESS**

**Data Sources**

* A data source name (DSN) is a data structure that contains the information about a specific database that an Open Database Connectivity (ODBC) driver needs in order to connect to it.
* ODBC database administrated it used to add, configure and delete datasources from the system.
* There are three kinds of DSN:
* User DSN
* System DSN
* File DSN

**User DSN**

* It is a user created for a specific user.
* A user can create a DSN for his account and no other user can use it or see it.
* If a connection is needed for a data sources they can use other own DSN to retrieve data.

**System DSN**

* It is a DSN that is seen by the entire system.
* Any user can see it as well as process it.

**File DSN**

* A file DSN is simply where the connection settings are returned to a file.
* The DSN for having a file DSN if we are in need to distribute a data sources connection to multiple users on different system without having to configure a DSN for each System.

**UNIT – V**

Introduction to. NET - Using Visual Studio – Stand alone application – web based applications,

Introduction to databases.

**Introduction to .Net**

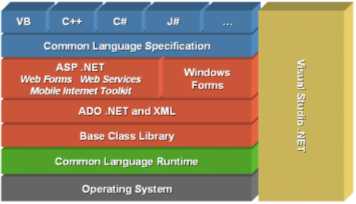
* .NET is an object oriented programming model
* It provides a platform independent framework
* It enables developers to quickly build and manage web applications, windows applications and web service applications
* It has a rich set of internet supporting languages.

**Advantages of .Net**

* Multiple Language Support
* Increased Performance
* Mobility support
* Flexible Data Access
* Effective Memory Management

**Components of .Net**

* The .NET Framework is made up of two major components:
* Common Language Runtime(CLR)
* Framework Class Library (FCL)

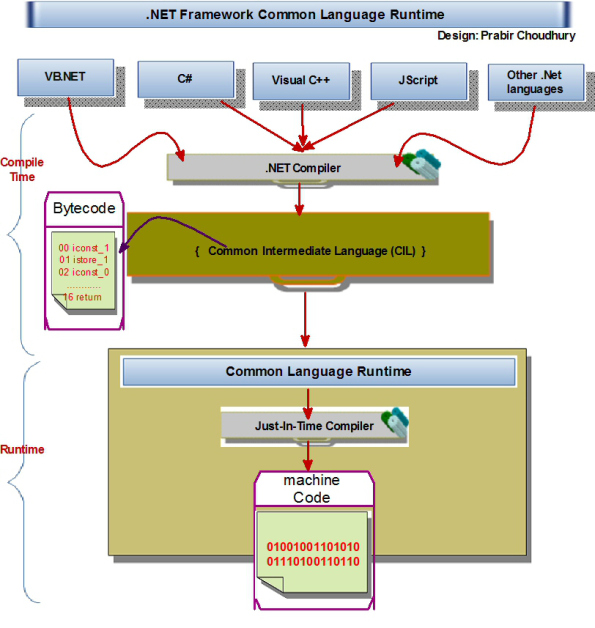


.Net framework hierarchy diagram

**CLR (Common Language Runtime)**

* It acts as the heart of the whole .NET frame work.
* CLR is the Runtime environment of the .NET frame work
* They manage the execution of code and provide services
* The process of CLR is
* Programmers write code in any .NET Compatible language such as C# , Visual Basic, Visual C++ for development.
* This part of process is called ‘Compile Time’
* .NET compiler converts the programmer’s source code into the ‘bytecode’ is called ‘Common Intermediate Language (CIL)’ or Intermediate Language (IL)

**CLR ARCHITECTURE**



The major component of CLR is,

* + CTS (Common Type System)
  + CLS (Common Language Specification)
  + CIL (Common Intermediate Language)
  + JIT (Just In Time) Compiler

**CTS (Common Type System)**

* It defines how types are declared, used and managed in runtime.
* It provides the data types, values, object types
* It helps developers to develop applications in different languages
* It means all the types used in applications share the same types defined under CLS.

**CLS (Common Language Specification)**

* It is a set of basic language features
* It solve the problem a sure on cross language access problem

**CIL (Common Intermediate Language)**

* The source code that is compiled to an intermediate code called as Common Intermediate Language

**JIT (Just In Time) Compiler**

* Just-In-Time compiler is to convert intermediate code to the machine code or native code

**Framework Class Library (FCL)**

* It is a collection of 2500 reusable classes, interfaces and value types
* It is divided into various namespaces based on the functionality
* The System namespace is the root namespace
* A namespace provides group of classes, interfaces, in-built functions etc.

**VISUAL STUDIO .NET**

* Microsoft Visual Studio is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](http://en.wikipedia.org/wiki/Microsoft).
* It can be used for developing console application, graphical user Interfaces, Windows forms, web services and web applications
* Visual Studio supports different [programming languages](http://en.wikipedia.org/wiki/Programming_language)

**Types of .NET**

* Window based application
* Web based application

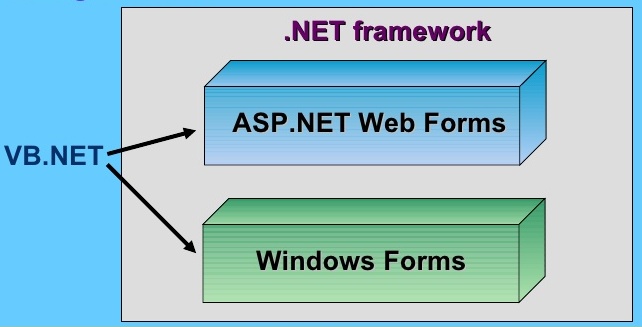
**Window based application**

* The window based application is also called as a standalone application
* The win form environment is the .NET frame work to build the window user interface application

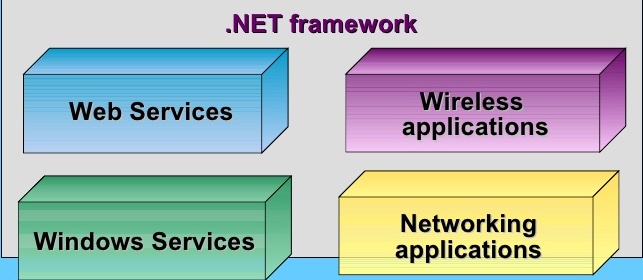
Example: VB.NET and C#.NET

**VB .NET**

* VB.Net has the ability to create stand alone applications that run locally on user’s computer
* Two main types of applications can be built using VB.Net

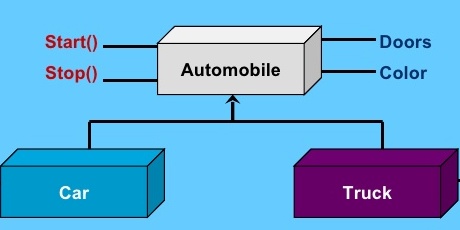


* Other types of applications can also built using VB.Net



**Features**

* VB.NET has the features of object oriented programming concepts like Inheritance, method overloading, exception handling etc
* Here inheritance allows developers to extend an existing code based to add new functionality
* Example



**Web based application**

* The collection of web pages is called as web application .
* The Web application is delivered using internet technology
* It is accessed by the user using a web browser like Internet Explorer, Google Chrome, and Mozilla Firefox etc.
* A collection of web pages, image, video etc., which gives relevant information, is called a web based application.

Example: ASP.NET

* A web application is any application that uses a web browser as a client. The 'client' is the application used to enter the information, and the 'server' is the application used to store the information.

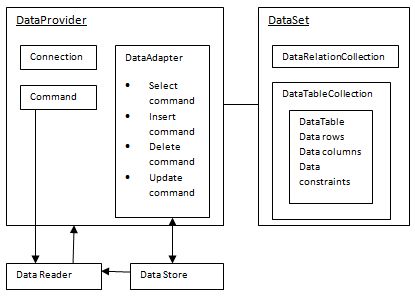
**Usage of Web application:**

* Web application makes us fast
* Application like electronic mailing (E-Mail), online shopping, online transaction, educational and entertainment information are the facilities provide by the web application.
* ASP.NET frame work provides facilities to create web based application

**Database access**

* Applications communicate with a database, to retrieve the data stored and update the database by inserting, modifying and deleting data
* Microsoft ActiveX Data Objects (ADO.Net) is a model used by the .Net applications for retrieving, accessing and updating data

ADO.Net Object Model



* The data residing in a data store or database is retrieved through the data provider
* An application **accesses** data either through a **dataset** or a **datareader**.
* **Datasets** store data in a disconnected cache and the application retrieves data from it.
* **Data readers** provide data to the application in a read-only and forward-only mode.
* **Data provider** is used for connecting to a database, executing commands and retrieving data, storing it in a dataset, reading the retrieved data and updating the database.
* ADO.Net consists 4 objects:

|  |  |
| --- | --- |
| **S.NO** | **Objects & Description** |
| 1 | **Connection** This component is used to set up a connection with a data source. |
| 2 | **Command** A command is a SQL statement or a stored procedure used to retrieve, insert, delete or modify data in a data source. |
| 3 | **DataReader** Data reader is used to retrieve data from a data source in a read-only and forward-only mode. |
| 4 | **DataAdapter** This is integral to the working of ADO.Net since data is transferred to and from a database through a data adapter. It retrieves data from a database into a dataset and updates the database. When changes are made to the dataset, the changes in the database are actually done by the data adapter. |

* There are following different types of data providers included in ADO.Net
* The .Net Framework data provider for SQL Server
* The .Net Framework data provider for OLE DB
* The .Net Framework data provider for ODBC
* The .Net Framework data provider for Oracle

**DataSet**

* DataSet is an in-memory representation of data
* When a connection is established with the database, the data adapter creates a dataset and stores data in it
* After the data is retrieved and stored in a dataset, the connection with the database is closed
* This is called the 'disconnected architecture'. The dataset works as a virtual database containing tables, rows, and columns
* The DataSet class is present in the System.Data namespace

**Connecting to a Database**

* The .Net Framework provides two types of Connection classes:
* SqlConnection - designed for connecting to Microsoft SQL Server.
* OleDbConnection - designed for connecting to a wide range of databases, like Microsoft Access and Oracle.

**Example :**

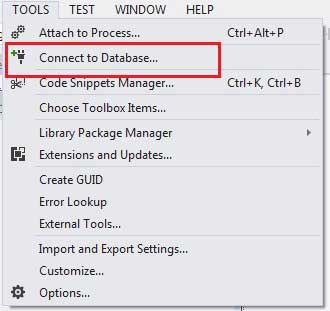
Database name: testDB

Table name: Customers

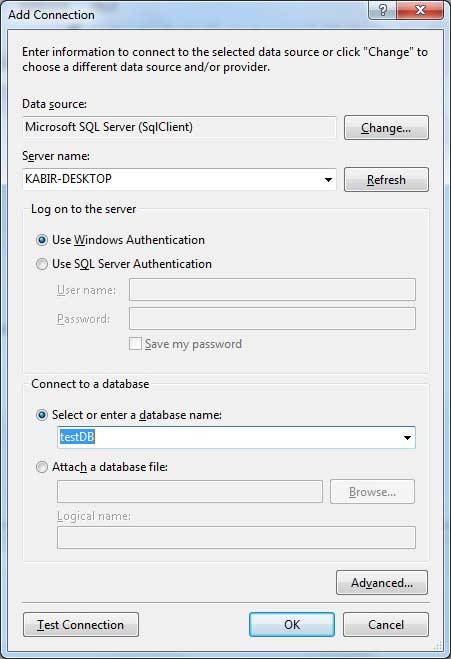
Data Source: Microsoft SQL Server

**Steps to connect the database**

1. Select TOOLS -> Connect to Database



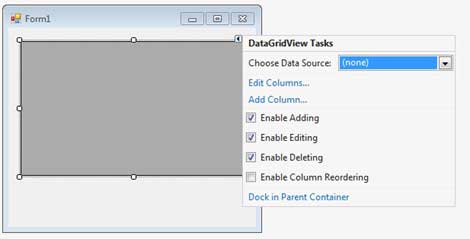
1. Select a server name and the database name in the Add Connection dialog box



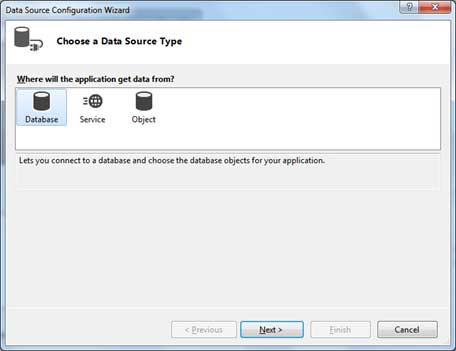
1. Click on the Test Connection button to check if the connection succeeded



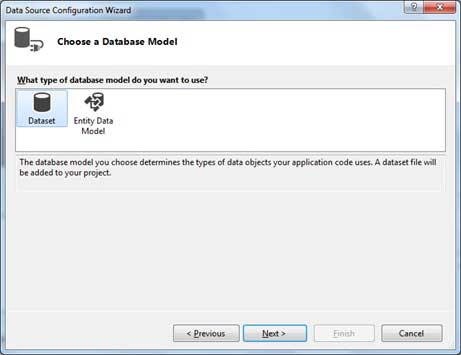
1. Add a DataGridView control on the form



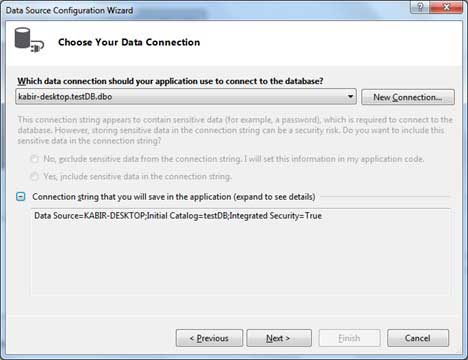
1. Click on the Choose Data Source combo box
2. Select Database as the data source type



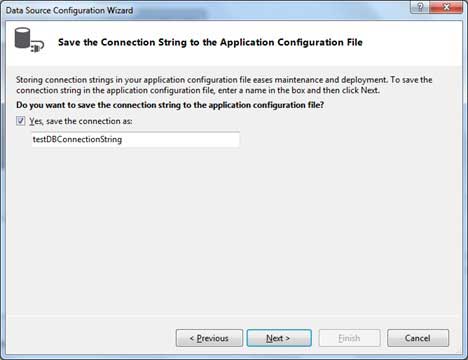
1. Choose DataSet as the database model



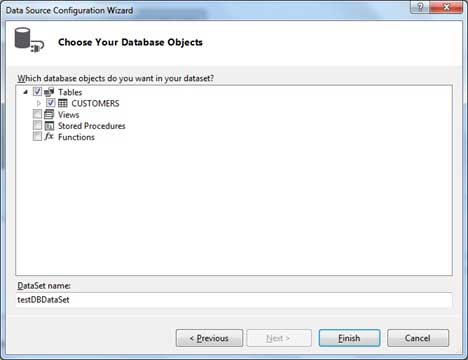
1. Choose the connection



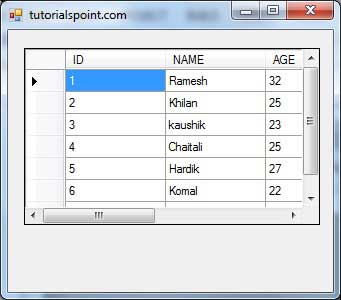
1. Save the connection string



1. Choose the table name Customers and click the Finish button



1. When the application is run using **Start** button it will show the output as,



**VALLUVAR COLLEGE OF SCIENCE AND MANAGEMENT,KARUR.**

**VISUAL PROGRAMMING**

**(II B.COM (CA) EVEN SEM)**

**PREPARED BY**

**Asst.Prof. DEPARTMENT OF COMMERCE**

**T.VIDHYA MCA,MBA(HR)**

**I-UNIT**

**1.What is FRAME?**

**Frames** into rectangular pixel blocks of different sizes, prediction of such blocks using previously reconstructed blocks, and transform coding of the residual signal. ... Inter **frames** are encoded with reference to previously encoded **frames** or reference buffers.

**2.What do you mean by forms?**

**Form** is the shape, visual appearance, or configuration of an object. In a wider sense, the **form** is the way something is or happens. **Form** may also refer to: **Form**(document), a document (printed or electronic) with spaces in which to write or enter data.

**3.What are the features of VISUAL BASIC?**

**Visual Basic** was designed to be a full-fledged programming language, complete with ordinary **features** like computation, string processing and more. It was integrated with a drag-and-drop approach to building user interfaces that would make it easy to use, even for novices or those strapped for time

**4.What is VISUAL BASIC?**

**Visual Basic** is a third-generation event-driven programming language from Microsoft for its Component Object Model (COM) programming model first released ...

**5.What do you mean by TITLE BAR?**

The **title bar** is typically used to display the name of the **application**, or the name of the open document, and may provide **title bar** buttons for minimizing, maximizing, closing or rolling up of **application** windows.

**6.Define a LIST BOX?**

A **List Box** is a control that provides a means of displaying a **list** of items (text, numbers, dates or whatever) on an **Access** form. Unlike the **Combo Box** it lacks a text **box** at the top in which the user can type.

**7.Any two functions of Properties Window?**

**The Properties window** is used to display **properties** for objects selected in **the two main** types of **windows** available in **the** Visual Studio integrated development environment (IDE). These **two** types of **windows** are: Tool **windows** such as Solution Explorer, Class View, and Object browser.

**8.What is Standard EXE?**

A **standard exe** application is one that is created using **Standard EXE** project. It is the most widely used Project type using VB6. **Standard EXE** application is normally the most widely used among the available Project types in Visual Basic. Stand-alone programs have an . **EXE** file extension.

**9.What is the meaning of Events?**

An **event** is something that happens, especially when it is unusual or important. You can use **events** to describe all the things that are happening in a particular situation. ... An **event** is a planned and organized occasion, for example a social gathering or a sports match.

**II UNIT**

**1.Write the meaning of Constant?**

something invariable or unchanging: such as. a : a number that has a fixed value in a given situation or universally or that is characteristic of some substance or instrument. b : a number that is assumed not to change value in a given mathematical discussion.

**2.Define Array?**

An **array** is a data structure that contains a group of elements. Typically these elements are all of the same data type, such as an integer or string. **Arrays** are commonly used in computer programs to organize data so that a related set of values can be easily sorted or searched.

**3.Different types of LOOPS in VB?**

**Loops** are of 2 **types**: entry-controlled and exit-controlled. 'C' programming provides us 1) while 2) do-while and 3) for **loop**. For and while **loop** is entry-controlled **loops**.

**4.Define variables and its DATATYPES?**

A **variable's** type determines **the** values that **the variable** can have and **the**operations that can be performed on it. For example, **the** declaration int count declares that count is an integer ( int ). ... Primitive types contain a single value and include types such as integer, floating point, character, and boolean.

**5.Give the role of Labels?**

Important functions of **labeling**: (i) Describe the Product and Specify its Contents: A**label** provides complete information regarding the product. It mainly includes ingredients of the product, its usage, and caution in use, cares to be taken while using it, date of manufacturing, batch number, etc.

**6.What are Frame Control?**

A **frame** is the instrument you use to package your power, authority, strength, information, and status. The moment your **frame** makes contact with the **frame** of another business person, they clash. If your **frame** wins this collision, you will enjoy**frame control**, where your ideas are accepted (and followed) by others.

**7.Define File System Control?**

**File system** or **filesystem** (often abbreviated to fs), **controls** how data is stored and retrieved. Without a **file system**, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stops and the next begins.

**8.List out the Application of Control Arrays?**

One **application of control arrays** is to hold menu items, as the shared event handler can be used for code common to all of the menu items in the **control array**.**Control arrays** are a convenient way to handle groups of **controls** that perform a similar function.

**9.Define a WHILE LOOP?**

**While Loop** – Do **While Loop**, Do Until **Loop**, **While** End **While** – VB2008. The Do. . . **Loop** executes a block of statements for as long as a condition is True, or until a condition becomes True. **Visual Basic** evaluates an expression (the **loop's**condition), and if it's True, the statements in the **loop's** body are executed.

**10.What is LOOP?**

In computer programming, a **loop** is a sequence of instruction s that is continually repeated until a certain condition is reached. Typically, a certain process is done, such as getting an item of data and changing it, and then some condition is checked such as whether a counter has reached a prescribed number.

**11.State the Explanation of if and else if?**

The **if**/**else** statement extends the **if** statement by specifying an action **if** the **if** (true/false expression) is false. With the **if**statement, a program will execute the true code block or do nothing.

**12. What are the different types of Arrays?**

* One dimensional (1-D) **arrays** or Linear **arrays**: In it each element is represented by a single subscript. The elements are stored in consecutive memory locations. ...
* Multi dimensional **arrays**: (a) Two dimensional (2-D) **arrays** or Matrix **arrays**: In it each element is represented by two subscripts.

**III UNIT**

Standard controls:

1. Define  **Controls.**

Special routines that would perform a specific tasks.

Eg: we have a control that accept data, displays pictures and drawlines etc.

2.Types of control.

Control name:

Toolbar control

Treeview control

Listview control

Imageview control

Common Dialog

StatusBar control

RichText control

Slider control

Grid control

3.Define **Form control**

1. The form is where the user interface is drawn.

2. The basic building block of vp is form.

3. User can add controls, Graphics and pictures to a form.

4. A form has many as 50 properties.

4.Define**Text box controls**

It is a standard control.

It is used to display or accept user Input in the form of text like name, customer id etc.The text box controls is also called on edit field or edit control.It displays the information entered by the user or assigned to the control in the code at run time.

5..Define Command button:

Create a button that user can choose to carry out a command.

The user will click on this button and the computer will perform the task associated with the button.

6. What are the Properties of command button?

1. Appearance

2. Caption

3. Font

4. Cancel

5. Height.

7. .Define Label box:

It allows you to display text that don't want to user to change.

It displays static text, titles and screen output from operations.

Properties of Label box:

1. Allignment - alligns caption within border.

2. Caption

3. Back color and fore color

4. Font

5. Border style.

8..Define Check box:

A checkbox is a control that allows the user to set (or) change the value of an item as true (or) false.

A check s also used to display multiple choices when the user can choose more than one.

When this empty squares is clicked, it gets marked by a check symbol.

Properties of Checkbox:

1 Caption

2. Font(type, style)

3. Value(Indicates checked (or) unchecked).

9..Define Frame control:

A frame is an object that outs as a container for controls.

A frame controls allows you to create a graphical or functional grouping controls.

To group controls, draw frame first and then draw controls inside the frame.

Properties of frame:

1. Caption

2. Font

10..Define Combo box:

A combo box controls combines the features of a text box and a list box.

The controls allows the user to select an item either by typing text into the combo box (or) by selecting it from the list.

A combo box is similar to list box.

Properties of combo box:

1. Appearance.

2. List

3. List count.

4. Style= 0(drop down combo)

Style= 1(simple combo)

Style= 2(drop down combo- user cannot change selections)

5.Text(most recently selected item)

11..Define List Box:

The list box controls displays a list of items from which user can select one or more item. If the number of items exceeds the number that can be displayed a scroll bar is

automatically added.

Properties of List box:

1. Appearances

2. List

3. List count

4. Multi select- 0- No Multiple selection allowed.

1- Multiple selection allowed

2- Group selection.

5. Text.

12..Define Radio button:

The radio button is also known as"option button".

It allows you to display multiple choices from which user can choose only one option.

The option may be like yes (OR) no, male(or) female and married (or) unmarried.

Properties of radio button:

1. caption

2. Font

3. Value- indicates of selected (or) not.

13.Define Image:

Image box is similar to picture box.

In that is allows you to place graphics information on a form.

Image boxes are more suited for static situation, that is cases where no modification will be done to the displayed graphics.

It display graphics in the following formates. Bit map, icon, metafile, JPEG (or) GIF, enhanced metafile.

Properties of Image:

1. pictures- Establish the graphics file to display in the image box.

2. Stretch- If false the Image box resize itself to fit the Graphics.

If true the Graphics the resize to fit the control area.

14.Define Picture box:

The picture box allows you to place graphics information on form.

It is best suited for dynamic environments.

The picture box behave very much like small forms with in a forms, processing most of the same properties as forms.

Properties of picture box:

1. Autosize- If true box adjust it a size to ft the displays graphics.

2. Font- Font size, style.

3. pictures- Establishes that grapics file to displays in the picture box.

15.Define Timer control:

Timer control response to the passage of time.

Many times, Especially in using graphics, we want to repeat certain operations at regular Intervals.

The timer tool allows such repetation. Timer also are useful for other kind of background processing.

Properties of timer control:

1. Enabled- If true the timer control is enable as soon as the form loads.

If false leave this property.

2. Interval- Number of milliseconds between timer events.

16.Define Timer Events:

The timer tool only has one Events, timer.

**IV UNIT**

1.What is File system?

VB provides three native tool box control for working with the file system.

1. Drive list box

2. Dir list box

3. File list box

2.Define DRIVE LIST BOX.

This is a drop down list box that will display the list of drives on your computer.

The user can select the valid directory from that.

It is used to displays a list of drive available in your computer.

When you place this control into form and run the program, you will be able to select different drive from your computer.

properties:

1. Drive- contains the name of currently selected drive.

3. Define Dirlist box.

The directory list box displays an ordered,hierarchical list of the user's disk directores and sub directories.

The directories structures is displayed in a list box..

Properties:

1.path-contains the current directory path.

4. Define File list box.

The file list box locates and lists files in the directory specified by ts path path property at run time.

You may select the types of files you want to display in the file list boxs.

Properties:

1. File name- contains the currently selected file name.

2. path- contains the current path directory.

3. pattern- contains a string that determines which file will be displayed.

5. What are the types of ACTIVEX CONTROL LIST?

1. Toolbar control

2. Treeview control

3. Listbox control

4. Imagelist control

5.Common dialog control

6.status bar control

7. richtext control

8. Slider control

9. Grid control

6. Define ACTIVEX CONTROL IN TOOLBAR.

By default, activex control doesn't add in tool bar.

To place, it click the right mouse button on the tool box and goto"components" select the controls and press "ok".

7.Define IMAGE LIST CONTROL.

It containns a collection of image that s used by other controls like list view, tree view and tool bar controls.

The controls uses bit map, cursor, con, JPEG (or) GIF files in the collection of list Image objects.

Place the Image list control onto the form.

set the property.

Add the image using Image list.

8.Define TREE VIEW CONTROL.

Tree view control is used to display the data in hierachical view.

It displays the drives,directories, subdirectories and fles in the forms of hierarchy.

9.Write about CREATING TREE VIEW CONTROL.

choose the treeview control from the toolbox and places it in the form.

Each item on tree vew is a node.

In tree view there is a parent bling and child node.

**V UNIT**

**1.Define DBMS?**

A database management system (**DBMS**) is a software package designed to **define**, manipulate, retrieve and manage data in a database. A **DBMS** generally manipulates the data itself, the data format, field names, record structure and file structure. It also**defines** rules to validate and manipulate this data.

**2.Who is a Client?**

**MIS**. Stands for "Management Information System." An **MIS** is a system designed to manage information within a company or organization. This includes employees, departments, projects, **clients**, finances, and other types of data.

**3.State the different types of Network?**

* Personal Area **Network** (PAN) ...
* Local Area **Network** (LAN) ...
* Wireless Local Area **Network** (WLAN) ...
* Campus Area **Network** (CAN) ...
* Metropolitan Area **Network** (MAN) ...
* Wide Area **Network** (WAN) ...
* Storage-Area **Network** (SAN) ...
* System-Area **Network** (also known as SAN)

**4.Define DataBase?**

A **database** is an organized collection of data, generally stored and accessed electronically from a computer system. ... The **database** management system (DBMS) is the software that interacts with end users, applications, and the **database**itself to capture and analyze the data.

**5.What is Hardware and Software?**

**Hardware** is the physical component of computing and includes technology like the keyboard, mouse, monitor, hard drive, etc. In Chapter 2, you looked at the **hardware** in a management information system. Without physical **hardware**, **software** cannot perform the function for which it is programmed.

**6.Explain the term Bridge in System?**

A **bridge** is a type of computer network device that provides interconnection with other **bridge** networks that use the same protocol. **Bridge** devices work at the data link layer of the Open **System** Interconnect (OSI) model, connecting two different networks together and providing communication between them.

**7.Define Bus Network?**

A **bus network** is an arrangement in a local area**network** (LAN) in which each node (workstation or other device) is connected to a main cable or link called the **bus**. The illustration shows a **bus network** with five nodes.

**II BCOM (CA) VISUAL PROGRAMMING**

**5 MARK QUESTIONS**

1.What is Integrated Development Environment?

2.Explain the Methods in Visual Basic

3.Write the Components of IDE

4. Write the applications of OLE

5.Explain the functions of Properties Window.

6.Mention the features of Visual Basic.

7.What are the Limitations of Array?

8.Describe about the variables functions.

9.Write the Syntax for Exit statement.

10.What is Select Case? Write its general Syntax.

11.Explain Scope of a Variable.

12.Distinguish between Objects and Controls.

13.Explain the Command button.

14.What is Status bar, Tool bar Control?

15.What are the uses of Check Box?

16.Explain the salient features of Combo box.

17.What is the Role of Label Control?

18.How does List view control differs from Tree view control.

19.What is Menu Editor?

20.Explain about the RICH Text box control.

21.Explain Common Dialog control.

22.What are various types of file system File System Control?

23.What is Data Grid? Explain it.

24.Write the features of .NET.

25.Write the various types of Data base.

26.Explain DLL.

27.What is meant by OLE? Explain it.

28.Write the Advantages of .NET.

29.Draft a note on Open Database Connectivity.

30.What do you mean by Active-x DDL?

**10 MARK QUESTIONS**

1.What is IDE? Explain the features of VB-IDE.

2.What are the elements of IDE?

3.How will you start a new Project with visual basic.

4.Explain the various types of visual basic editions.

5.Explain the different types of LOOP’s in Visual Basic.

6.Describe for...... next statement.

7.Write note on i)Image control ii)Picture box

8.What is Rich Text box control? How do you change the colour of the selected text.

9.Discuss on Common dialog boxes.

10.Write a detailed note on Combo box.

11.Describe about file system in Visual Programming.

12.Discuss about Active-x Control.

13.What are the advantages of Data base?

14.Explain ODBC.

15.What is .NET? Explain its Features.