Diploma in
Information Technology Application

DITA

Study Material

Youth Computer Training Centre

Topics Covered:

- Programming Concepts – Algorithm & Flow Chart
- Visual Basic
- Overview of VB.Net
- RDBMS concepts with MS-Access
- Internet & E-mail

Contents:

Chapter 1: Algorithm & Flowchart

Introduction

An algorithm is a finite sequence of well-defined instructions to solve a problem or perform a specific task. It is independent of any programming language and focuses purely on the logical steps needed to achieve the desired outcome.

Characteristics of an Algorithm

- Input: Takes zero or more inputs.
- Output: Produces at least one output.
- Definiteness: Each step must be clear and unambiguous.
- Effectiveness: Each step must be basic enough to be executed.
- Finiteness: Must terminate after a finite number of steps.

Steps to Create an Algorithm

- Understand the Problem: Analyze the requirements and constraints.
- Define Inputs and Outputs: Determine what data is required and what results are expected.
- Break down the Problem: Divide the problem into smaller manageable tasks.
- Write Logical Steps: Define a sequence of clear and actionable steps to achieve the result.
- Test the Algorithm: Verify its correctness by simulating it with test cases.

Example of an Algorithm

Problem: Find the largest of three numbers.

Algorithm:

```
1.    Start.
2.    Input three numbers: A, B, C.
3.    If A > B and A > C, then A is the largest.
4.    Else if B > A and B > C, then B is the largest.
5.    Else, C is the largest.
6.    Print the largest number.
7.    Stop.
```

Advantages of Algorithms

- Clarity: Provides a clear set of steps to solve a problem.
- Efficiency: Identifies the optimal way to perform tasks.
- Language-Independent: Can be implemented in any programming language.
- Error Reduction: Helps debug and test logical errors before implementation.

Applications of Algorithms

1. Programming: Forms the foundation of coding logic.
2. Data Processing: Sorting, searching, and data analysis.
3. Optimization: Solving mathematical and engineering problems efficiently.
4. Automation: Driving decision-making in AI, robotics, and other fields.

Types of Algorithms

1. Brute Force: Tries all possible solutions until the correct one is found.
   Example: Linear search.
2. Divide and Conquer: Breaks the problem into smaller sub problems, solves them independently, and combines the results.
   Example: Merge sort, quicksort.
3. Greedy Algorithm: Makes the most optimal choice at each step.
   Example: Dijkstra's shortest path algorithm.
4. Dynamic Programming: Breaks the problem into overlapping sub problems and solves them efficiently.
   Example: Fibonacci sequence, knapsack problem.
5. Backtracking: Explores all possible solutions and abandons solutions that fail to satisfy the constraints.
   Example: Sudoku solver.

Importance of Algorithms

1. Problem Solving: Helps in breaking down and solving problems logically.
2. Optimization: Ensures tasks are performed in the least time and with the least resources.
3. Scalability: Algorithms form the backbone of systems that handle large-scale data efficiently.

   Mastering algorithms is key to efficient programming and effective problem-solving!

Flowchart

A flowchart is a visual representation of a process or algorithm. It uses standardized symbols to describe the flow of steps in a task, making it easier to understand and communicate the logic or sequence of operations.

Features of a Flowchart

1. Visual Representation: Provides a clear and graphical illustration of a process.
2. Logical Sequence: Shows the steps in a process in a logical order.
3. Decision Points: Includes conditional branches for decision-making.

Common Flowchart Symbols

Symbol	Name	Purpose
Oval	Start/End	Indicates the beginning or end of a process.
Rectangle	Process	Represents an action or operation to perform.
Diamond	Decision	Represents a decision point with options.
Parallelogram	Input/output	Represents input or output operations.
Arrow	Flow line	Shows the direction of the flow.

Steps to Create a Flowchart

a) Define the Problem: Identify what the flowchart will solve or explain.
b) List the Steps: Write down all the actions and decisions in sequence.
c) Choose Symbols: Use appropriate symbols for each type of action or decision.
d) Draw the Flowchart: Connect the symbols in the correct sequence with arrows.
e) Verify: Ensure that flowchart accurately represents the process.

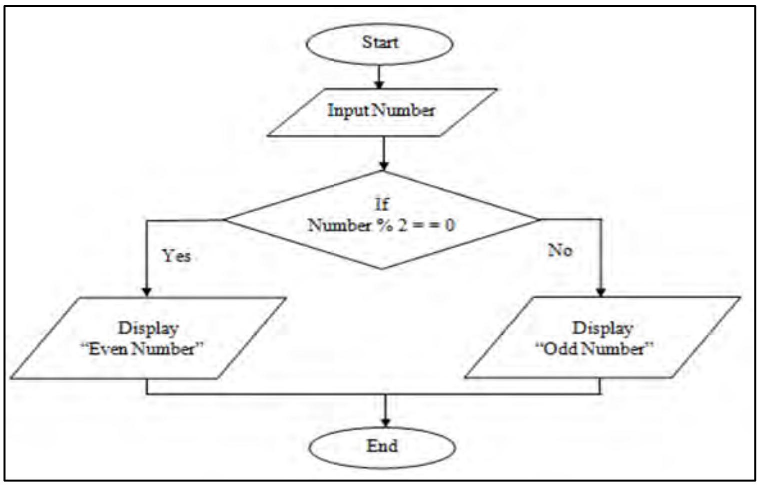
Example of a Flowchart

Problem: Check if a number is even or odd.

Steps:

1) Start.
2) Input a number N.
3) Check if N % 2 == 0.
    i) If true, print "Even".
    ii) If false, print "Odd".
4) End.

Flowchart:



Advantages of Flowcharts

1. Improves Understanding: Makes complex processes easy to understand.
2. Facilitates Communication: Useful for explaining processes to non-technical stakeholders.
3. Error Identification: Helps spot logical errors before implementation.
4. Documentation: Serves as a reference for process execution and maintenance.

Applications of Flowcharts

1. Programming: Visualize the logic before writing code.
2. Process Management: Document workflows in business operations.
3. Education: Explain concepts in a structured and visual way.
4. Problem Solving: Break down problems into manageable steps.

Types of Flowcharts

1. System Flowchart: Represents the flow of data in a system.
2. Process Flowchart: Describes a business process.
3. Program Flowchart: Details the logic of a programming task.

Flowchart vs Algorithm

Aspect	**Algorithm**	**Flowchart**
Definition	Step-by-step instructions to solve a task.	Graphical representation of a process.
Representation	Written in text or pseudocode.	Uses symbols and arrows.
Usage	Focuses on the logic of the process.	Focuses on visualization of the process.

Summary:

A flowchart is a powerful tool to visualize processes, making it easier to design, communicate, and debug systems. Its simplicity and clarity make it a universal method to explain logic and workflows across industries.

Check your Understanding

Question 1

What is an algorithm?

a) A flowchart
b) Step by step instructions used to solve a problem
c) A flowchart or pseudocode
d) A decision

Question 2

A process box is _______ in shape.

a) Connector
b) Rectangular
c) Oval
d) Circle

Question 3

Part of an algorithm which is repeated for fixed number of times is classified as

a) iteration
b) selection
c) sequence
d) reverse action

Question 4

Diamond shaped symbol is used in flowcharts to show the

e) decision box
f) statement box
g) error box
h) if-statement box

Question 5

What is a list of instructions in a proper order to solve a problem called?

a) Flowchart
b) Sequence
c) Algorithm
d) None of these

Question 6

Symbol used in flowchart such as rectangle with horizontal lines on two sides is used for

a) rhombus
b) parallelogram
c) circle
d) trapezoid

Question 7

In a flowchart a calculation (process) is represented by

a) A rectangle
b) A rhombus
c) A parallelogram
d) A circle

Question 8

A process box is _______ in shape.

a) Connector
b) Rectangular
c) Oval
d) Circle

Question 9

Algorithm and Flow chart help us to

a) Know the memory capacity
b) Identify the base of a number system
c) Direct the output to a printer
d) Specify the problem completely and clearly

Question 10

Diamond shaped symbol is used in flowcharts to show the

a) decision box
b) statement box
c) error box
d) if-statement box

Chapter 2: Visual Basic

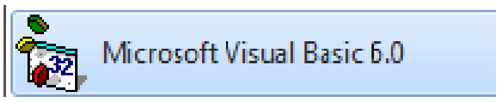
Introduction

Welcome to Microsoft Visual Basic, the fastest and easiest way to create applications for Microsoft Windows®. Whether you are an experienced professional or brand new to Windows programming, Visual Basic provides you with a complete set of tools to simplify rapid application development.

About Visual Basic ?

The "Visual" part refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, you simply add or arrange objects on the screen. If you've ever used a drawing program such as Paint, you already have most of the skills necessary to create an effective user interface.

The "Basic" part refers to the BASIC (Beginner's All-Purpose Symbolic Instruction Code) language, which has been used by more programmers than any other language in the history of computing. Visual Basic has evolved from the original BASIC language and now includes several hundred statements, functions, and keywords, many of which relate directly to the Windows GUI.



Data access features allow you to create databases, front-end applications, and scalable server-side components for most popular database formats, including Microsoft SQL Server and other enterprise-level databases.

ActiveX™ technologies allow you to use the functionality provided by other applications, such as the Microsoft Word processor, Microsoft Excel spreadsheet, and other Windows applications. You can even automate applications and objects created using the Professional or Enterprise editions of Visual Basic.

Internet capabilities make it easy to provide access to documents and applications across the Internet or intranet from within your application, or to create Internet server applications.

Your finished application is a true .exe file that uses the Visual Basic Virtual Machine, which you can freely distribute.

Visual Basic Editions

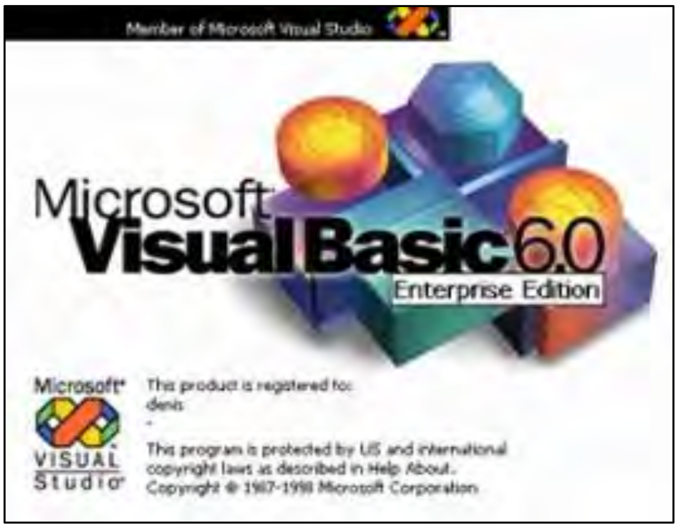
Visual Basic is available in three versions, each geared to meet a specific set of development requirements.

The Visual Basic Learning Edition allows programmers to easily create powerful applications for Microsoft Windows and Windows NT®. It includes all intrinsic controls, plus grid, tab, and data- bound controls.

The Professional Edition provides computer professionals with a full-featured set of tools for developing solutions for others. It includes all the features of the Learning Edition, plus additional ActiveX controls, the Internet

Information Server Application Designer, integrated Visual Database Tools and Data Environment, Active Data Objects, and the Dynamic HTML Page Designer.

The Enterprise Edition allows professionals to create robust, distributed applications in a team setting. It includes all the features of the Professional Edition, plus Back Office tools such as SQL Server, Microsoft Transaction Server, Internet Information Server, Visual SourceSafe, SNA Server, and more.

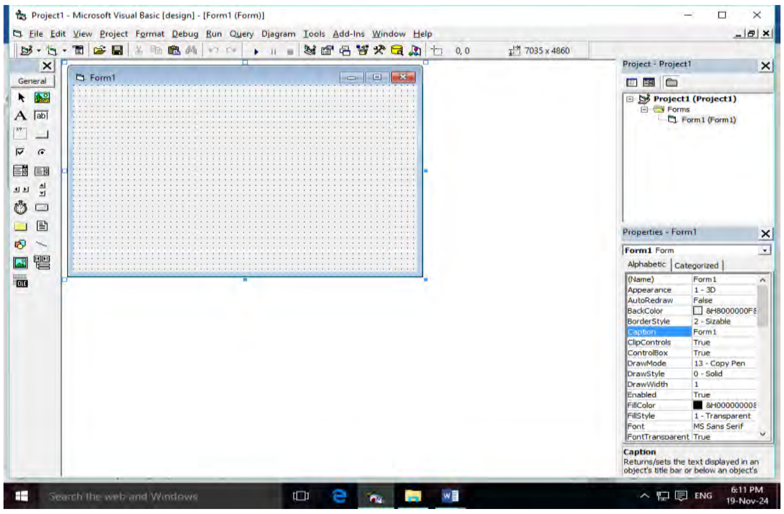

Member of Microsoft Visual Studio
Microsoft Visual Basic 6.0
Enterprise Edition
Microsoft Visual Studio
This product is registered to: denis
This program is protected by US and international copyright laws as described in Help About.
Copyright © 1987-1998 Microsoft Corporation

Starting Visual Basic IDE

When you run the Visual Basic Setup program, it allows you to place the program items in an existing program group or create a new program group with new program items for Visual Basic in Windows. Once that's done, you're ready to start Visual Basic from Windows.

To start Visual Basic from Windows

1. Click Start on the Task bar.
2. Select Programs, and then Microsoft Visual Basic 6.0


Project1 - Microsoft Visual Basic [design] - [Form1 (Form)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
General
Form1
Project - Project1
Project1 (Project1)
Forms
Form1 (Form1)
Properties - Form1
Form1 Form
Alphabetic
Categorized

(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000
FillStyle	1 - Transparent
Font	MS Sans Serif
FontTransparent	True

Caption
Returns/sets the text displayed in an object's title bar or below an object's

When you first start Visual Basic, you'll see the interface of the integrated development environment. If you want to save this interface, click File → Save.

In the saving process,

Visual Basic first saves the form (with the .FRM extension).

Finally, Visual Basic saves the project (with the .VBP extension).

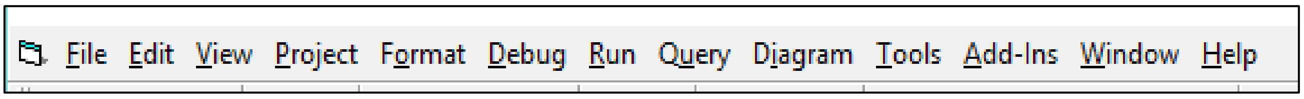
In Visual Basic, you can work on three different mode.

They are –

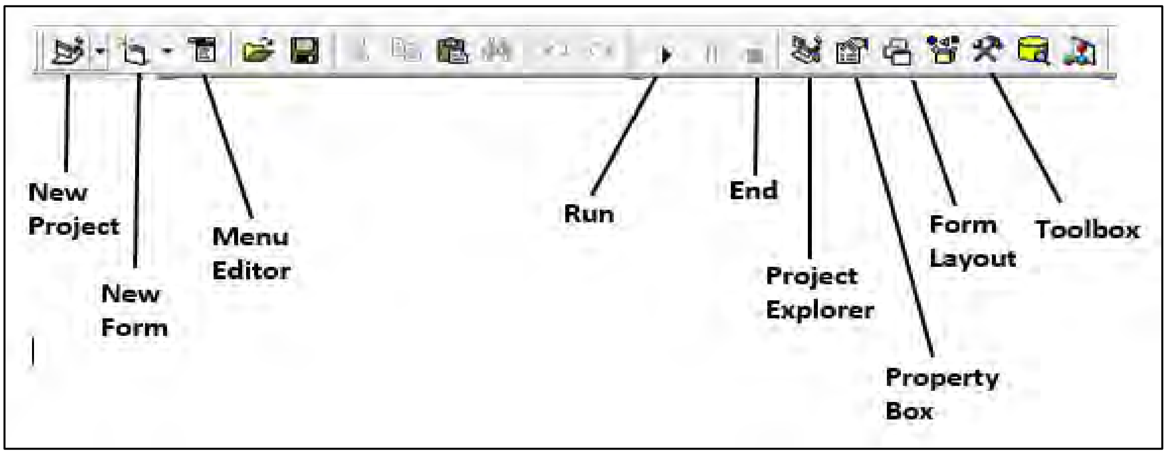
1) Design Mode – The mode during which you build an application in the development environment by adding controls, setting control or form properties, and so on. In contrast, during run mode, you interact with the application like a user.

2) Run Mode – This mode works when you execute any Visual Basic program. For this, you can press F5 key from keyboard or can click on Start Option under Run Menu.

Menu Bar



Toolbars

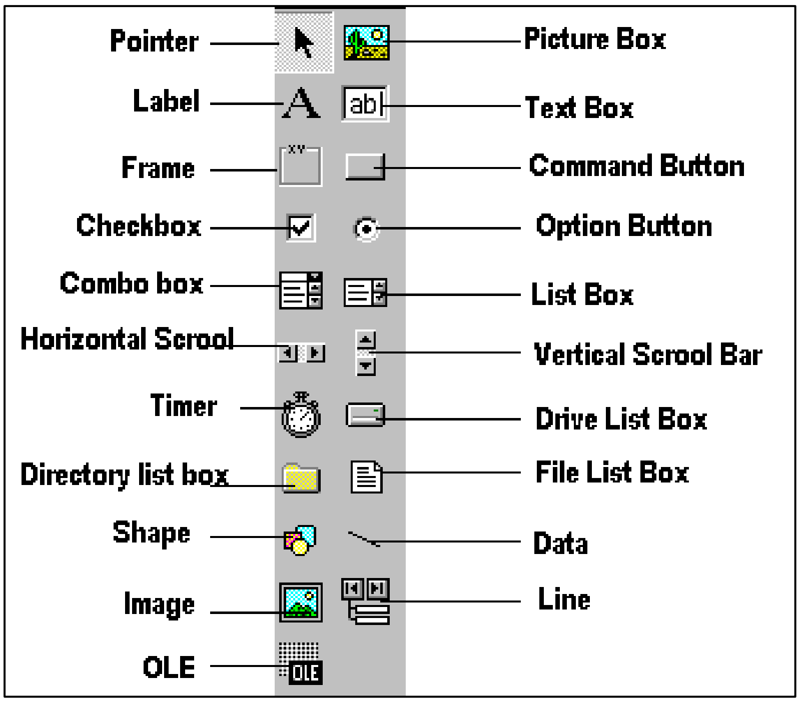


Toolbox

The Visual Basic toolbox contains the tools you use to draw controls on your forms. There are three broad categories of controls in Visual Basic:

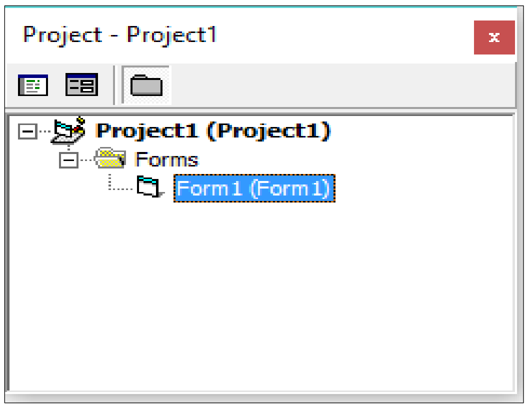
1. Intrinsic controls, such as the command button and frame controls, are contained inside the Visual Basic .exe file. These controls are always included in the toolbox, unlike ActiveX controls and insertable objects, which can be added or removed from the toolbox.

2. ActiveX controls exist as separate files with a .ocx file extension. These include controls available in all editions of Visual Basic (such as RichTextBox, Common Dialog Control, and so on), as well as those available only in the Professional and Enterprise editions (such as ListView, Toolbar, Animation, and Tabbed Dialog). Many third-party ActiveX controls are also available.
3. Insertable Objects, such as a Microsoft Excel Worksheet object containing a list of all your company's employees, or a Microsoft Project Calendar object containing the scheduling information for a project. Since these can be added to the toolbox, they can be considered controls. Some of these objects also support Automation (formerly called OLE Automation), which allows you to program another application's objects from within a Visual Basic application. See "Programming with Objects," for more information on Automation.



Project Explorer Window

The Project Explorer appears. This window allows you to organize the parts of your program into folders for easy manipulation, as all parts of the project are displayed in the Project Explorer in a tree view.

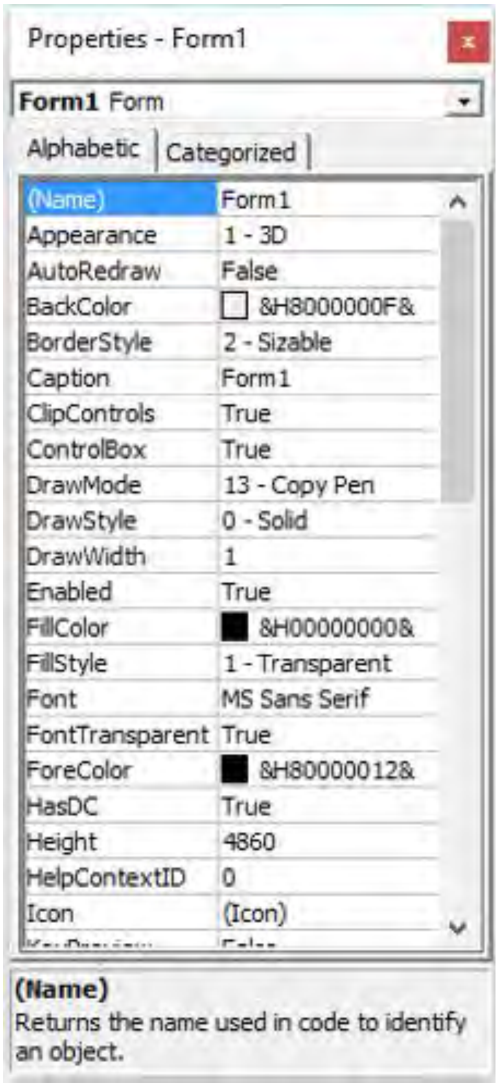


The Project Explorer can be very useful when working on a larger project, especially when the IDE is filled with design and code windows. To focus on a specific part of the project, simply find it in the Project Explorer and double-click it. Doing so brings the appropriate window to the foreground, and if you've clicked on a form, it will open that form in the Properties window.

You can also add and remove items by right clicking them with the mouse in the Project Explorer. You can remove forms by right –clicking them and selecting the Remove item in the pop – up menu that appears or you can save them to disk or switch between the form's code window and the form itself.

Properties Window

The Properties window appears. This is where you set an object's properties; for example, you can set the caption of command buttons, the text in textboxes, and literally hundreds of other properties.



(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	☐ &H8000000F&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	■ &H00000000&
FillStyle	1 - Transparent
Font	MS Sans Serif
FontTransparent	True
ForeColor	■ &H80000012&
HasDC	True
Height	4860
HelpContextID	0
Icon	(Icon)

(Name)
Returns the name used in code to identify an object.

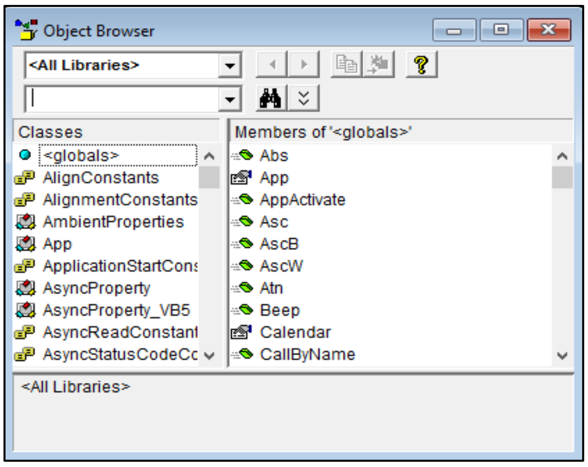
When you select an object, such as a control, in Visual Basic using the mouse, that object's properties appear in the Properties window. To change or examine a property setting, simply locate the property in the window (properties appear on the left, with their settings on the right).

It's important to note that there are two types of properties in Visual Basic: design-time properties and runtime properties. The properties that appear in the Properties window are design-time properties.

You can open the Properties window by pressing the F4 key on the keyboard or selecting it from the View menu.

Object Browser

Displays the classes, properties, methods, events, constants, and procedures available in your project, as well as those from packages and object libraries. You can use it to find and utilize objects you create, as well as objects from other applications.



The contents of the Object Browser are application-specific. When an application is loaded into memory, it may populate the browser with specific contents. As a result, the contents may differ from what you saw during your last session. Once you create a project or select a command that uses a particular application, the Object Browser will display contents specific to that application and keep them available as long as the application remains in memory. Common items are always available.

You can obtain help for the Object Browser by pressing F2 or by choosing Object Browser help from the View menu. Now, you'd immediately understand the problem because Visual Basic would display an error message for the incorrectly spelled "TemVal." Since the Option Explicit statement helps you catch these types of errors, it's a good idea to use it in all your code.

Different Data Types of Variables

You can declare variables in several ways. Most often, you use the Dim statement to declare a variable. If you do not specify the variable type when using Dim, it creates a variant, which can function as any type of variable. In addition to Dim, you can use ReDim to resize space for dynamic arrays. You can also use Private to restrict the variable to a module or form, Public to make it global (accessible to all modules or forms), or Static to ensure its value doesn't change between procedure calls.

These different methods of declaring variables are summarized in the table below.

Keyword	Does This
Dim	Using Dim alone creates variants. Use the As keyword to specify variable type.
Private	Makes variable available only in the current form/module
Public	Makes variable global – variable is available to the rest of program
ReDim	Reallocates storage space for dynamic array variables
Static	Variable preserve the value between procedure calls
Type	Declares a user type

The possible variable types and their ranges are shown in the following table –

Variable Type	Bytes of Storage
Boolean	2
Byte	1
Currency	8
Date	8
Decimal	12
Double	8
Integer	2
Long	4
Object	4
Single	4
String	N/A

Using Form Events

After adding the necessary forms to your project, you must determine which event to use. Forms are code modules that contain procedures called events, which respond to system or user input by running the corresponding code. One example is the Click event, which runs whenever the mouse is clicked on the form. When an event executes its code, this action is often referred to as firing or triggering the event.

You can use many types of form events. Depending on how a form is used, some of these events may not fire, while others will always fire. Form events are generally triggered in the following order:

Managing Forms Control

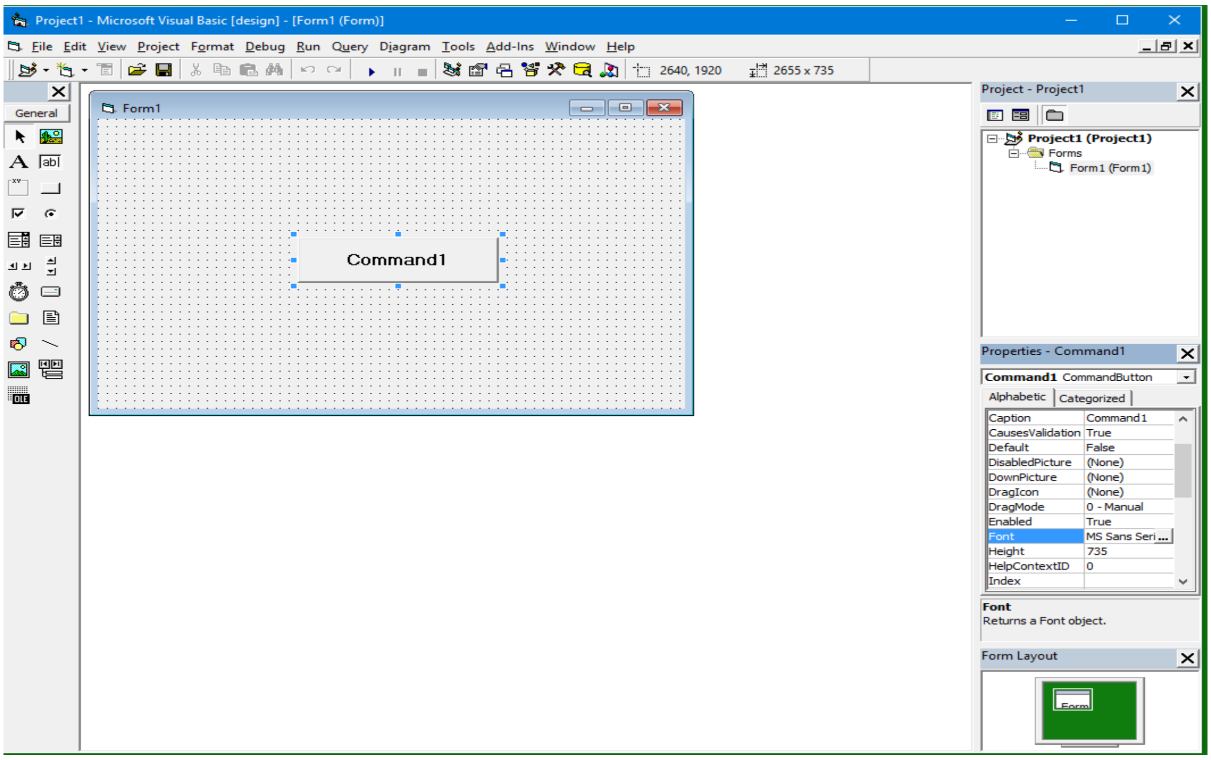
In this chapter, we'll take a look at handling forms in Visual Basic. There's a great deal to see about form handling and we'll look at it all.

The Parts of a Form

Forms are the names for windows in Visual Basic (originally, you called windows under design forms and the actual result when running a window, but common usage has named both forms now), and you add controls to forms in the Integrated Development Environment (IDE).

When designing a form in the Visual Basic IDE, you can see several aspects of the form. At the top is the title bar, which displays the form's title; here, that's "Form1." On the right side of the title bar is the control box, which includes the minimize, maximize, and close buttons. These are common controls that users expect in most windows, though they may be inappropriate in certain cases, such as in dialog boxes.

The main area of a form – the area where everything takes place – is called the client area. In general, Visual Basic code work with controls in the client area and leaves the rest of the form to Visual Basic (in fact, the client area is itself a window). Finally, the whole form is surrounded by a border, and there are several types of borders that you can use.


Project1 - Microsoft Visual Basic [design] - [Form1 (Form)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
2640, 1920
2655 x 735
General
Form1
Command1
Project - Project1
Project1 (Project1)
Forms
Form1 (Form1)
Form Layout

Properties - Command1

Command1 CommandButton

Alphabetic | Categorized

Caption	Command1
CausesValidation	True
Default	False
DisabledPicture	(None)
DownPicture	(None)
DragIcon	(None)
DragMode	0 - Manual
Enabled	True
Font	MS Sans Seri
Height	735
HelpContextID	0
Index	

Font
Returns a Font object.

Setting Title Bar Text

When you run the above Visual Basic Project, then at run mode, the Title of the Program will Display as Form1. But nobody can like this. Actually, this stymies a lot of Visual Basic programmers, because the text in the title bar seems like something that Windows itself messages, not the program. In fact, it's up to the program and setting the text in the title bar couldn't be easier. At design time, you just change the form's Caption Property from the property box.

You can also set the Caption property at run time in code like this –

```
Private Sub Form_Load()

Form1.Caption = “First Visual Basic Program”

End Sub
```

Add/Remove Min/Max Button and Setting Window Border

Forms usually come with minimizing and maximizing buttons, as well as a close box at the upper right. However, that’s not appropriate in all cases. To remove this button, you can set the form’s Control Box property to false. Note that that the usual buttons are missing from the form at the upper right.You can also set what buttons are in a form by setting its border type. For example, if you set the border style to a fixed type, the minimizing and maximizing buttons are disappeared.

ScaleHeight to 1000 : To draw a scatter plot of your data, then you could use Pset() to set individuals pixels directly. If one of the points to graph was (233, 599), you could draw that dot this way – Pset(233,599).

Loading, Showing and Hiding Forms

There are times when you might want to work with a form before displaying it on the screen, such as initializing it with graphics and other elements. In such cases, you can load the form into memory using the Load statement.

You don’t need to explicitly load or unload forms to show or hide them – the loading and unloading processes are automatic. Forms are typically loaded explicitly only if you need to work on them before displaying them, as Visual Basic recommends. However, you don’t actually need to use the Load statement for this, because simply referring to a form causes Visual Basic to load it automatically. This means you can use the Show() or Hide() methods without manually loading the form first.

To display a form on the screen, you use the Show() method. Here’s an example where we load a new form, Form2, and then show it:

```
 Private Sub Command1_Click()

  Load Form2

  Form2.Show

End Sub
```

After displaying a form, you can hide it using the Hide() method and unload it (although this is not always necessary) with the Unload statement. You typically unload forms when you have many open and are concerned about memory usage. Here’s an example where we hide Form2 and then unload it :

Private Sub Command1_Click()

Form2.Hide

Unload Form2

End Sub

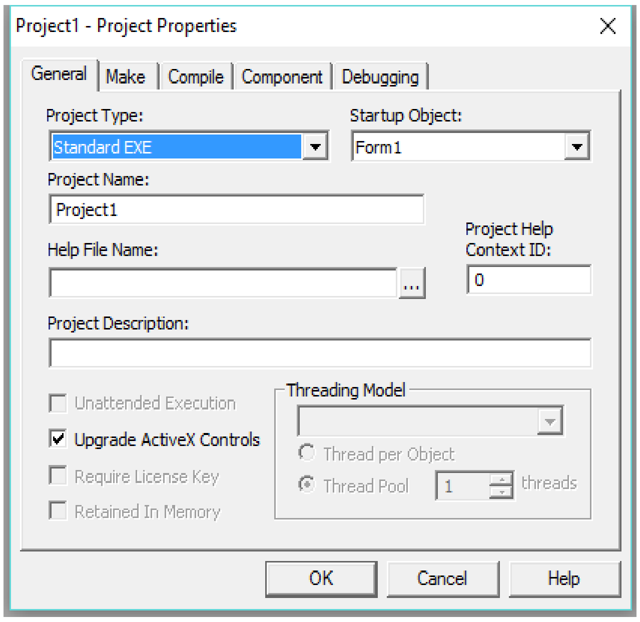
Setting the Startup Form

Suppose you've developed a Visual Basic application with multiple forms. (To add a new form, click on Project → Add Form.) In your application, the second form is the opening form. However, when you run the project, Form1 is the first form displayed.

You can set the startup form following these steps:

Select the Project → Properties Item.

In the Project Properties window that opens, select the General tab, as shown.



Project1 - Project Properties

General | Make | Compile | Component | Debugging

Project Type: Standard EXE

Startup Object: Form1

Project Name: Project1

Help File Name:

Project Help Context ID: 0

Project Description:

☐ Unattended Execution
☑ Upgrade ActiveX Controls
☐ Require License Key
☐ Retained In Memory

Threading Model
○ Thread per Object
● Thread Pool 1 threads

OK | Cancel | Help

All about Message Boxes and Input Boxes

Visual Basic provides two ways of displaying message boxes and input boxes : using MsgBox() and

InputBox(). We'll cover their syntax in the following subsections.

The MsgBox() Function

You use MsgBox() to display a message to the user and get a return value corresponding to one of the buttons in the message box. Here's the syntax –

Msgbox (prompt, buttons, title, helpfile, context)

The prompt arguments holds the string displayed as the message in the dialog box. The maximum length of prompt is approximately 1024 characters. If prompt is made up of more than one line, you can separate the lines using a carriage return characters (chr(13)), a linefeed character (chr(10)), or both (chr(13) & chr(10)) between each line.

The InputBox() Function

In Visual Basic 6.0 (VB6), the InputBox function is used to prompt the user for input through a dialog box. The user can type a response, which is then returned as a string.

Syntax: InputBox (Prompt, [Title], [Default], [X], [Y], [HelpFile], [Context])

- Prompt (Required): The message displayed in the dialog box, which tells the user what information they should provide.
- Title (Optional): The title for the dialog box. If omitted, the title is "Input".
- Default (Optional): The default text that appears in the input field. If omitted, the field is empty.
- X (Optional): The horizontal position of the input box. If omitted, the box appears in the center of the screen.
- Y (Optional): The vertical position of the input box. If omitted, the box appears in the center of the screen.
- HelpFile (Optional): The name of a help file that provides help for the InputBox.
- Context (Optional): A numeric context ID that links the help file to the dialog box.

Example:

```
Dim UserInput As String

UserInput = InputBox ("Please enter your name:", "User Input", "John Doe")


If UserInput <> "" Then

   MsgBox "Hello, " & UserInput & "!"

Else

   MsgBox "No input provided."

End If
```

Setting's A Button's Caption

You use a button's Caption property to set its caption. This property is available at both design time and runtime.

After you add a button to a form, you set its caption by placing the appropriate text in the Caption property in the Properties window. You can also change the button's caption at runtime, of course.

As an example, we'll use the following program -

```
Dim xNow As Boolean

Private Sub Command1_Click()

 If xNow = True Then

Command1.Caption="X"
Else

Command1.Caption = "O"

        End If

xNow = Not xNow

        End Sub

Private Sub Form_Load()

xNow = True

        End Sub
```

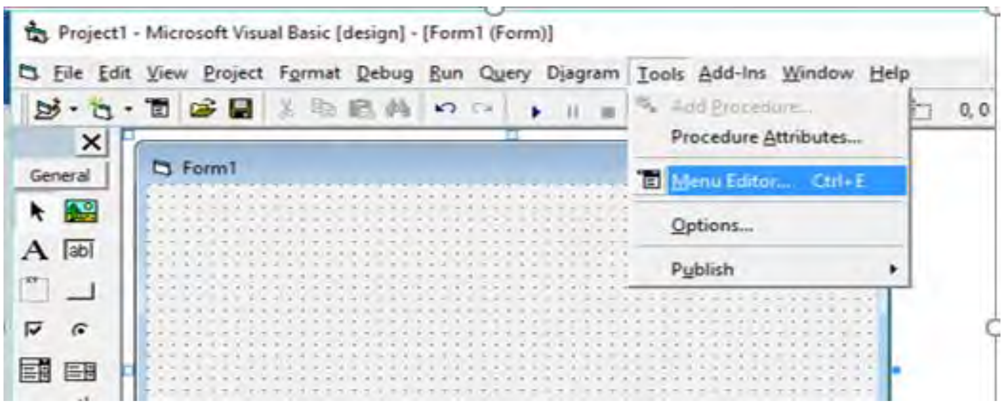
Creating the Menu with the Menu Editor

The Menu Editor is the only tool in Visual Basic for creating or modifying a menu in design mode. Although you can make some changes to the menu at runtime, the vast majority of the menu is created in the Menu Editor. You can access the Menu Editor by clicking the Menu Editor button on the toolbar, choosing the Menu Editor option from the Tools menu, or pressing Ctrl+E. Once you have opened the Menu Editor, you can begin creating the items for your menu. For each item, you must specify two key properties: the Name and Caption. The Caption property identifies the menu item to the user, while the Name property identifies the menu item to your program. The first item you create is a main menu bar item.

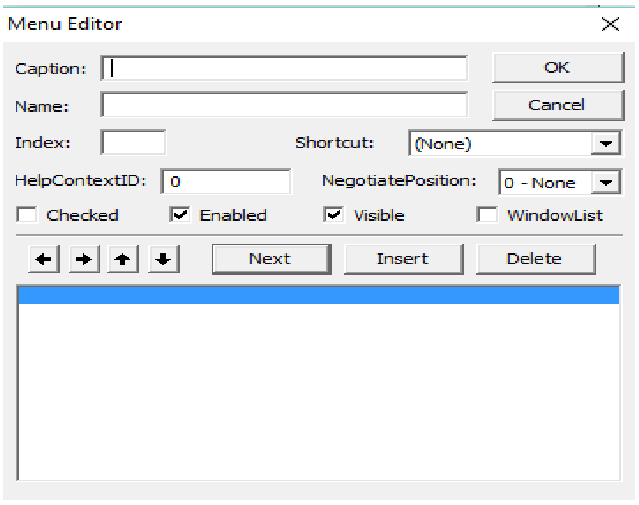
The only property you are required to specify for a menu item is the Name property. However, even if you plan to specify the menu captions using code, it is a good idea to set the Caption property for each menu item to assist with program design.

To add a menu to a form:

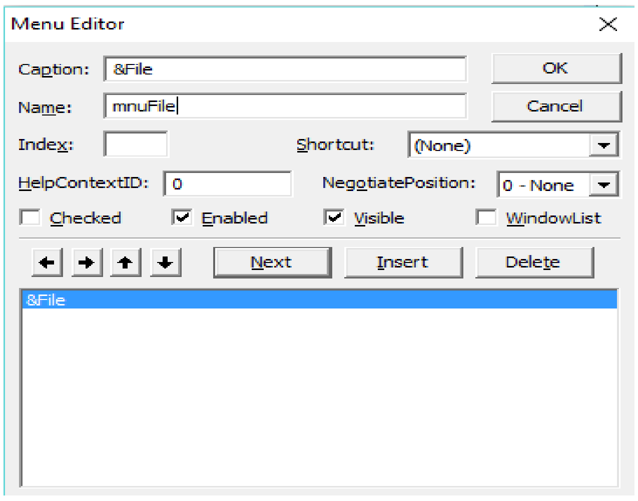
On the Tools menu, click the Menu Editor option.


Project1 - Microsoft Visual Basic [design] - [Form1 (Form)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Add Procedure...
Procedure Attributes...
Menu Editor... Ctrl+E
Options...
Publish
General
Form1
0, 0

If a child window has a menu, its menu replace the parent's menu when the child form has the focus.



Caption:
Name:
Index:
Shortcut: (None)
HelpContextID: 0
NegotiatePosition: 0 - None
Checked ☐ Enabled ✓ Visible ✓ WindowList ☐

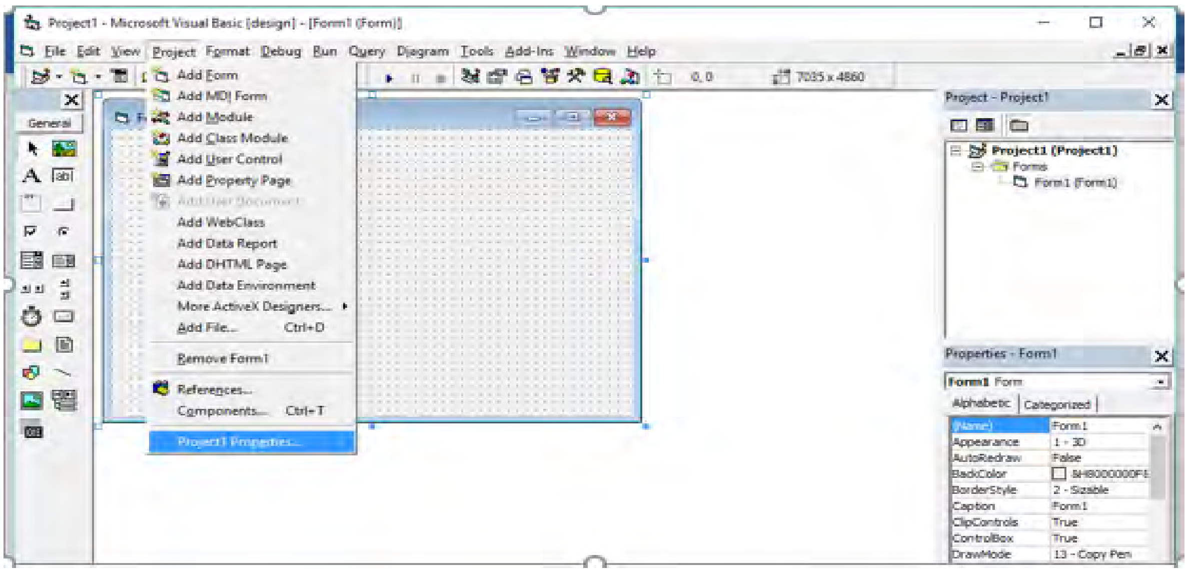


Caption: &File
Name: mnuFile
Index:
Shortcut: (None)
HelpContextID: 0
NegotiatePosition: 0 - None
Checked ☐ Enabled ✓ Visible ✓ WindowList ☐

&File

Making the MDI Parent the Startup Form

In an MDI application, you usually want the MDI parent form to be the startup form. To do this, go to the Project menu and select the <projectname> Properties item, where <projectname> is the name of your project. After selecting this menu item, the Project Properties dialog box will appear.



(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

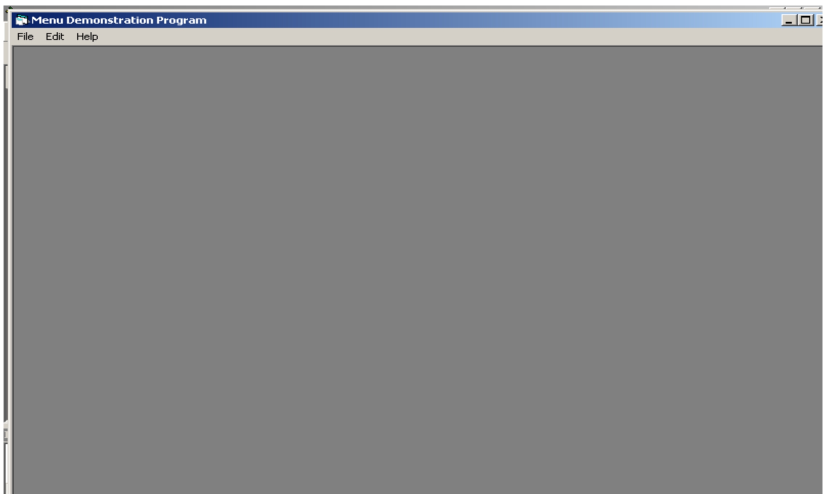
Step -1 Project <projectname> Properties

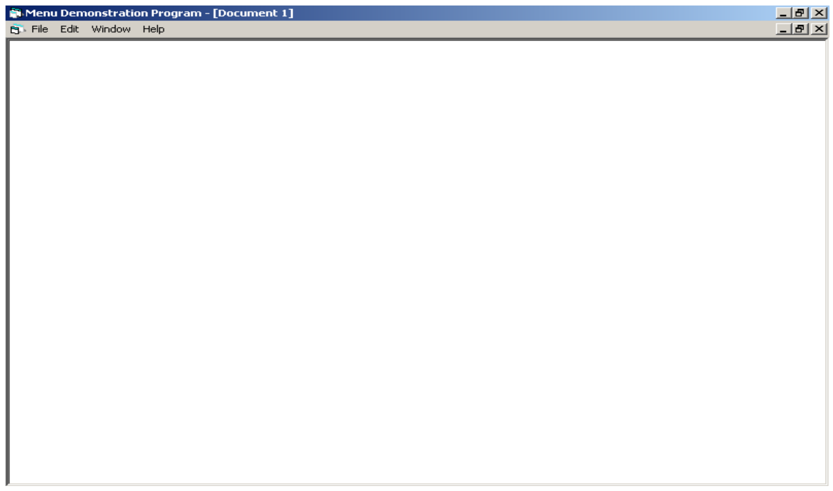
On the first tab, there is a 'Startup Object' drop-down selector. Select the name of your MDI parent form as the startup form object. Now, when you run your application, the MDI parent form will appear automatically

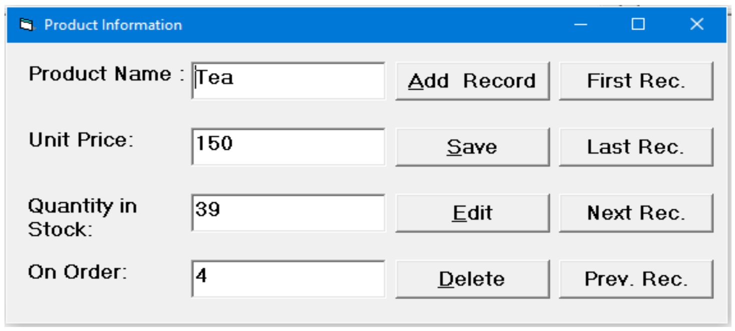
If the count is 2, implying that only the MDI form and the current document are open, the code sets the Visible property of the mnuEdit and mnuWindow menu items to False in order to hide them.

```
    Private Sub FrmNotepad_Unload()
    If Forms.Count =2 Then
      mdiMain.mnuEdit.Visible = False
       mdiMain.mnuWindow.Visible = False
       End if
  End Sub
```

You can now run the program and watch how the menus are displayed and hidden as documents are opened or closed.







Product Information

Product Name :	Tea	Add Record	First Rec.
Unit Price:	150	Save	Last Rec.
Quantity in Stock:	39	Edit	Next Rec.
On Order:	4	Delete	Prev. Rec.

Add, Save, Modify, and Delete Command buttons

2) Now add the following code in the Click event of cmdAdd procedure:

```
Dim control AS Object
For Each Control In Form1
If TypeOf Control Is TextBox Then
 Control.Text = “” txtProdName.SetFocus
rst.AddNew
End If
Next
```

3) Add the following Code in the Click event of cmdSave procedure:

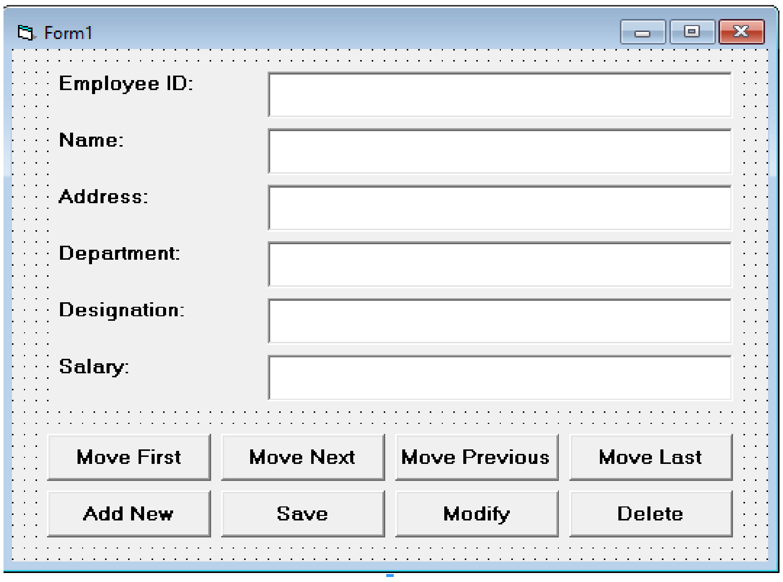
```
 rst!ProductName = txtProdName.Text
rst!UnitPrice = txtUnitPrice.Text
rst!UnitsInStock = txtInStock.Text
rst!UnitsOnOrder = txtOnOrder.Text
rst.Update
```

4) Add the following code in the Click event of the cmdModify procedure:

```
rst!ProductName = txtProdName.Text
rst!UnitPrice = txtUnitPrice.Text
rst!UnitsInStock = txtInStock.Text
rst!UnitsOnOrder = txtOnOrder.Text
rst.Save
```

5) Add the following Code in the Click event of the cmdDelete procedure:

```
Dim I as Integer
I = MsgBox(“Are you sure you want to Delete this recod?”,vbYesNo)
        If I = vbNo Then Exit Sub
        rst.Delete
        With rst
        .MoveNext
        If .EOF Then
        .MoveLast
        End With        ShowData
```



Task II Reference the ADODC

Select the Components option from Project Menu Check the Microsoft ADO Data Control 6.0 (OLE DB) OK

Task III

Draw the ADODC on the form and set its properties

Task IV

Set the two properties (Datasource and DataField) of Textboxes to display information.

To navigate and interact with the database using ADODC following methods of the RecordSet Object are used

Task	Method	Syntax
Adding New Record	AddNew	Adodc1.Recordset.AddNew
Saving Record	Update	Adodc1.Recordset.Update
Deleting Record	Delete	Adodc1.Recordset.Delete
Moving First Record	MoveFirst	Adodc1.Recordset.MoveFirst
Move to last Record	MoveLast	Adodc1.Recordset.MoveLast
Move to Next record	MoveNext	Adodc1.Recordset.MoveNext
Move To previous Record	MovePrevious	Adodc1.Recordset.MovePrevious

Code to implement All the task

Option Explicit Dim obj as Object

Private Sub cmdFirst_Click() Adodc1.Recordset.MoveFirst

 End Sub

Private Sub Form_Load()

Adodc1.Visible = False

```
For Each obj In Form1
If TypeOf obj Is TextBox Then
 obj.Enabled = False
End If
  Next
cmdsave.Enabled = False
End Sub
```

Programming with ADO

Visual Basic provides a variety of ways to accomplish any given task, and creating database applications is no exception. You've seen how to create a database application using the ActiveX Data Objects Data control (ADODC). With the ADODC, you set up the data control and then bind standard controls, such as text boxes, to it to display and edit the data. This approach allows you to quickly and easily create an interface for viewing and modifying data.

Another approach to creating database applications is to use the ActiveX Data Objects (ADO) in code. This method requires more effort than using the ADODC, but there are some advantages to using pure code. Some of these advantages are:

You can more easily validate all the information entered by the user before it is saved to the database.

You reduce the possibility of locking conflicts in multi-user systems because your program controls when the record is locked.

You can use transaction processing to speed up data storage and to help preserve data integrity.

You can create database applications that do not require a visual interface. The data control is only good for handling programs that have a visual component.

Actually, using ADO in code and using the ADODC are not all that different. The ADODC is actually a wrapper around the ADO objects. For example, when you click one of the navigation buttons of the ADODC, you are invoking one of the Move methods of ADO. Also, when you used code to add and delete records with the ADODC, you were actually writing ADO code.

This chapter shows you how to create the various ActiveX Data Objects, how to retrieve information using the objects, and how to store new or changed information to the database

Understanding the ActiveX Data Objects

The ActiveX Data Objects model provides Visual Basic with a robust environment for creating database applications. Each of these objects contains method and properties that control their behavior and enable them to perform certain data manipulation tasks. The ADO objects are designed to work equally well with Microsoft Jet databases and SQL databases such as SQL Server or Oracle. ADO also works with almost any Open Database Connectivity (ODBC) compatible database.

Many objects are contained within the ActiveX Data Objects. However, you will look closely at three of them in this chapter. These objects are:

Connection: This object provides the link between your programs and a data source. The data source can be a Jet database, on ODBC database, or a SQL Server data source. When you create the Connection object, you are performing the same function that the ADODC does when you set the ConnectionString property. Each Connection objects can support multiple lower-level objects, such as recordsets or commands, and other objects. The Connection object is also where transaction processing is handled in the ADO model.

Rich Text Boxes

Rich Textboxes were designed to be a step beyond plain text, and because many word processors allow you to save text in this format, they can provide a link between different types of word processors. Using RTF boxes, you can also create your own simple word processors, which is exactly what the Visual Basic Application Wizard does when you create an application with it. You'll find that the child windows in an Application Wizard program have a rich text box stretched across them, ready for the user to work with

Adding An RTF Box to a Form

So you've decided to make the move from Text Boxes to rich text boxes and for this you have to add a rich text box to a form, follow these steps:-

1. Select the Project ® Components menu items.
2. Click the Controls tab in the Components Box.
3. Find and select the Microsoft Rich Textbox Control 6.0 box and click on Ok to close the Components box.
4. The Rich Textbox control now appears in the toolbox and you can use it to add rich text boxes to your forms.

Accessing Text in a Rich Text Box

To access text in a rich text box, you can use two properties : Text and TextRTF. As their names imply, Text holds the text in a rich text box in plain text format (like a text box) and TextRTF holds the text in Rich Text Format.

Here's an example where we read the text in RichTextBox1 without any RTF codes and display that text as plain text in RichTextBox2.

```
Private Sub Command1_Click()

RichTextBox2.Text=RichTextBox1.Text

End Sub
```

Here's the same operation where we transfer the text including all RTF codes that is, here we're transferring rich text from one rich text box to another.

```
Private Sub Command1_Click()

RichTextBox2.TextRTF=RichTextBox1.TextRTF

End Sub
```

Selecting Text In Rich Text Boxes

Rich text boxes support the SetText property just like standard text boxes. However, SetText only works with plain text. You can set the start and end of plain text selection with the SelStart and SelLength properties.

If you want to work with RTF selected text, on the other hand, use the SelRTF property. For example, here's how we select the first 10 characters in RichTextBox1 and transfer them to RichTextBox2 using SelRTF.

```
Private Sub Command1_Click()

RichTextBox1.SelStart=0

RichTextBox1.SelLength=10

RichTextBox2.TextRTF=RichTextBox1.SelRTF

End Sub
```

Using Bold, Italic, Underline and StrikeThru

To make text bold, italic, underline or strikethrough, you use the Selbold, SelItalic, SelUnderline and SelStrikethru properties. These properties work on selected RTF text only, so you have to select the text whose format you want to change.

To make this clearer, here's an example where we set the underline, bold, italic and strikethrough properties of text. We start by placing some text into a rich text box.

Private Sub Form_Load()

```
RichTextBox1.Text = "This Rich Text Supports Underline, Bold, Italic and Strikethru Text"

End Sub
```

Private Sub Command1_Click()

```
RichTextBox1.SelStart = 24

RichTextBox1.SelLength = 9

RichTextBox1.SetFocus

RichTextBox1.SelUnderline = True

RichTextBox1.SelStart = 35

RichTextBox1.SelLength = 4

RichTextBox1.SetFocus

RichTextBox1.SelBold = True

RichTextBox1.SelStart = 41

RichTextBox1.SelLength = 6

RichTextBox1.SetFocus
```

RichTextBox1.SelItalic = True

RichTextBox1.SelStart = 52

RichTextBox1.SelLength = 10

RichTextBox1.SetFocus

RichTextBox1.SelStrikeThru = True

End Sub

Setting Fonts and Font Sizes in Rich Text Boxes

In a Rich Text Box, if you want the Selection's Font, you just set the SetFontName to the new font names (For example, Arial or Times New Roman). To set a selection's font size, you just set the SelFontSize property.

Here's an example.

Private Sub Command1_Click()

RichTextBox1.SelStart = 0

RichTextBox1.SelLength = Len(RichTextBox1.Text)

RichTextBox1.SelFontName = "Arial Black"

RichTextBox1.SelFontSize = 20

End Sub

Indenting Text In Rich Text Boxes

One of the aspects of word processors that users have gotten used to is the ability to indent text and rich text boxes (which are designed to be RTF word processors in a control) have this capability. To indent paragraph – by – paragraph you use these properties (you set them to numeric values to indicate the indentation amount, using the measurement units of the underlying form, which is usually twips.)

- SelIndent – Indents the first line of the paragraph.
- SelHangingIndent – Indents all other lines of the paragraph with respect to SelIndent.
- SelRightIndent – Sets the right indentation of the paragraph.

To use these properties on a paragraph of text, you either select the paragraph (using SelStart and SelLength).

Private Sub Command1_Click()

RichTextBox1.SelIndent = 500

RichTextBox1.SelHangingIndent = -250

RichTextBox1.SelRightIndent = 100

End Sub

Common Dialog Control

In this section, we will examine the Windows Common Dialogs, which provide a powerful and professional set of dialog boxes for interacting with the user. Microsoft created the Common Dialogs to promote a consistent user interface across all Windows programs. In fact, the Common Dialogs work well—they make programming easier for developers.

The Common Dialog control can display five different dialog boxes: Open a File, Save a File, Set a Color, Set a Font, and Print a Document.

Creating and displaying a Windows Common Dialog

Adding a Windows Common Dialog control to your program is easy : just follow these steps :-

1. Select the Project menu and click Components Menu item.
2. Select the Controls Tab in the Components box that opens.
3. Select the entry labeled Microsoft Common Dialog Control 6.0, then click on OK to Close the Components tab.
4. Add a Common Dialog Control to a form in the usual way – just double click the Common Dialog Tool in the toolbox or select it and paint the control on the form.
5. Add the code you want to open the dialog box and make use of values the user sets.

To display various dialog boxes, you use these Common Dialog methods (For example, Common Dialog1.ShowColor).

- **ShowOpen – Show Open Dialog Box.**
- **ShowSave – Show Save As Dialog Box.**
- **ShowColor – Show Color Dialog Box.**
- **ShowFont – Show Font Dialog Box.**
- **ShowPrinter – Show Print or Print Options Dialog Box.**

You can also set the Common Dialog's Action property to do the same thing. Here are the values you can place in the Action property.

- 0 – No Action
- 1 – Displays the Open Dialog Box.
- 2 – Displays the Save As Dialog box.
- 3 – Displays the Color Dialog Box.
- 4 – Displays the Font Dialog Box.
- 5 – Displays the Print Dialog Box.

Now that you've added a Common Dialog Control to your program, refer to the individual topics in this chapter for the dialog box you want to work with to see how to retrieve values from the user.

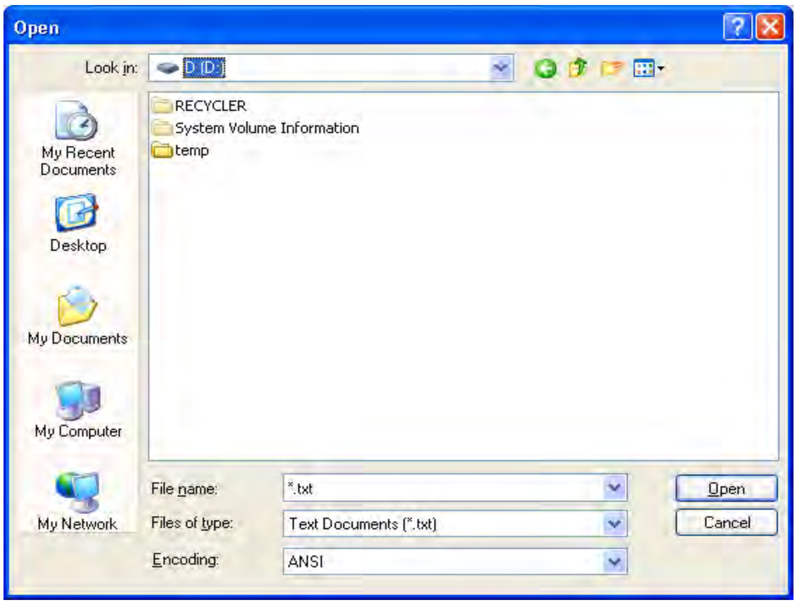
Setting A Common Dialog's Title

Although some programmers may question the wisdom of changing a Common Dialog's Title, you can do it using the DialogTitle property. As an example, here we're changing the title of an Open Dialog Box to "Select a File to Open".

Private Sub Command1_Click()

CommonDialog1.DialogTitle = "Select a File to Open" CommonDialog1.ShowOpen

End Sub



RECYCLER
System Volume Information
temp

File name: *.txt
Files of type: Text Documents (*.txt)
Encoding: ANSI

Open Dialog Box with New Title

Using Open and Save Dialog Boxes

Probably the most common use of the Common Dialog Control is to Display File Open and File Save As Dialog boxes, and you display those dialog boxes with the Common Dialog control's ShowOpen and ShowSave methods. These methods need no arguments passed to them – to set various options, you set the Common Dialog Control's Flags property.

You can also set the Filter property so the dialog box display only certain types of files, such as text files.

To find out what file the user wants to work with, you check the Common Dialog's FileName property after the user clicks on OK in the dialog box.

Following section display the code of File Open Button.

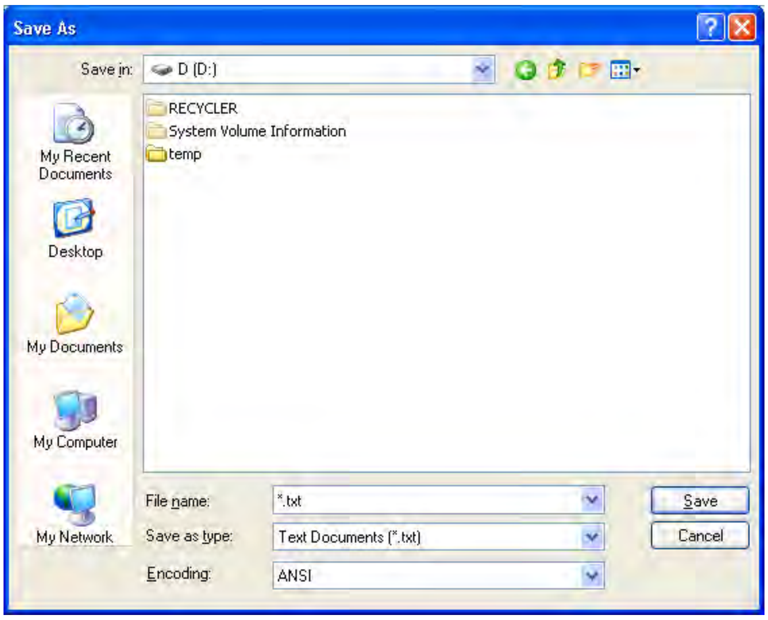
Private Sub Command1_Click()

CommonDialog1.Filter="Text Files(*.txt)|*.txt|Image File(*.jpg,*.gif)|*.jpg;*.gif"

CommonDialog1.ShowOpen

MsgBox "File to Open - " & CommonDialog1.FileName

End Sub



File name: *.txt

Save as type: Text Documents (*.txt)

Encoding: ANSI

The following section display the code for Save As Button

Private Sub Command1_Click()

CommonDialog1.DefaultExt = "txt" CommonDialog1.ShowSave

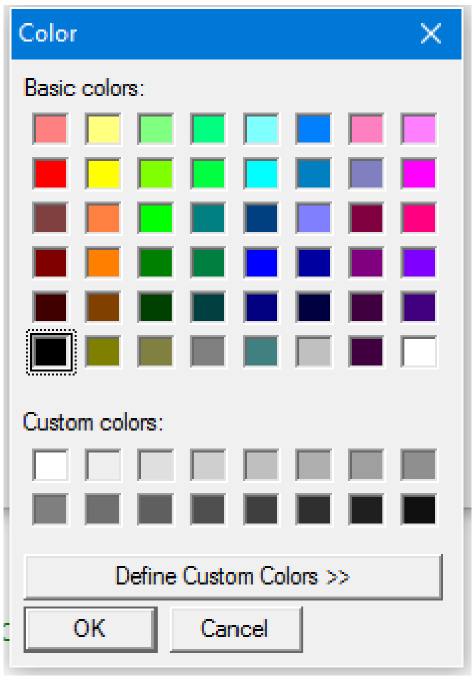
MsgBox "File to Save : " & CommonDialog1.FileName

End Sub

Using Color Dialog Box

To let the user select colors,

you use the Color Dialog Box, and you display that dialog box with the Common Dialog method ShowColor. To retrieve the color the user selected, you use the dialog box's Color property. There are special flags you can set for the Color Dialog box– see the next topic for more information.



```
Private Sub Command1_Click()
CommonDialog1.ShowColor
Form1.BackColor = CommonDialog1.Color
  End Sub
```

Using a Font Dialog Box

You use the Common Dialog Control's ShowFont method to show a Font dialog box. Note that before you use the ShowFont method, you must set the Flags property of the Common Dialog Control to one of three constants to indicate if you want to display screen fonts, printers fonts or both. The possible values are as follows :-

- cdlCDScreenFonts – show screen fonts
- cdlCFPrinterFonts – show printer fonts
- cdlCFBoth – show both types of fonts.

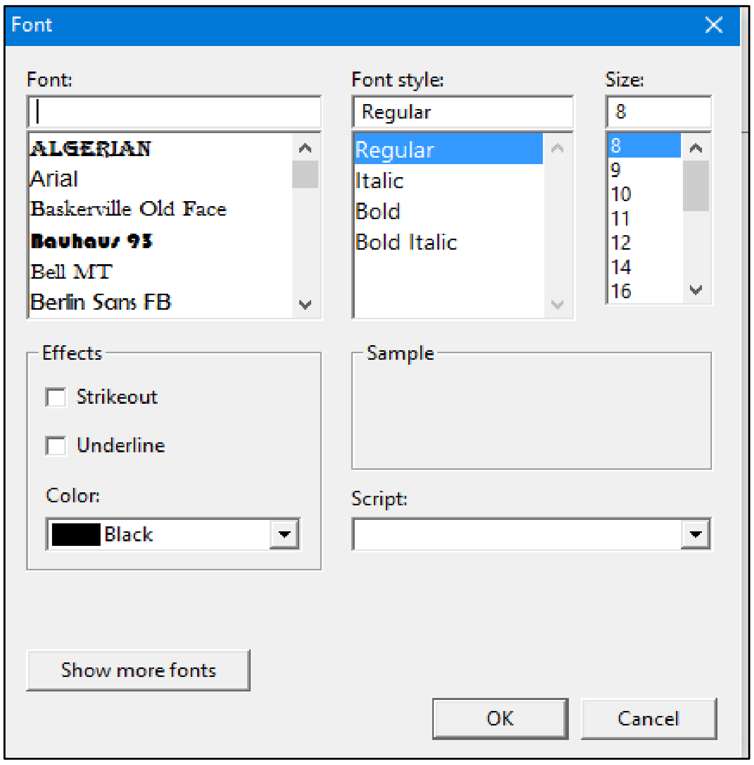
If you don't set one of these in the Flags property, a message box is displayed advising the user that "There are no fonts installed", which will probably cause them to panic. To see more possible settings for the Flags property, take a look at the next topic in this chapter.

When the user dismisses the Font Dialog box by clicking on OK, you can determine their font selections using these properties of the Common Dialog Control.

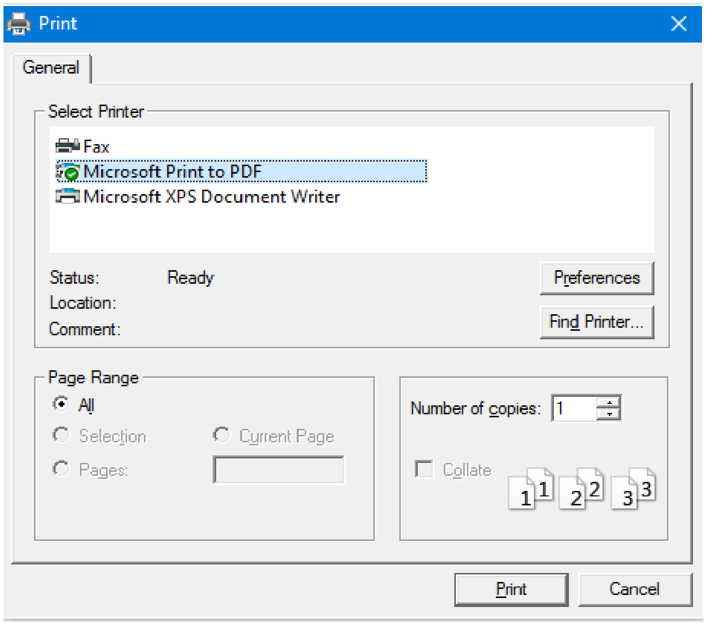
- Color – The selected Color. To use this property, you must first set the Flags property to cdlCFEffects.
- FontBold – True if bold was selected.
- FontItalic – True if italic was selected
- FontStrikethru – True if strikethrough was selected. To use this property, you must first set the Flags property to cdlCFEffects.

- FontUnderline – True if underline was selected. To use this property, you must first set the Flags property to cdlCFEffects.
- FontName – The selected font name
- FontSize – The selected font size

```
Private Sub Command1_Click()

CommonDialog1.Flags = cdlCFBoth Or cdlCFEffects

CommonDialog1.ShowFont

Text1.FontName = CommonDialog1.FontName

Text1.FontBold = CommonDialog1.FontBold

Text1.FontItalic = CommonDialog1.FontItalic

Text1.FontUnderline = CommonDialog1.FontUnderline

Text1.FontSize = CommonDialog1.FontSize

Text1.FontStrikethru = CommonDialog1.FontStrikethru

Text1.ForeColor = CommonDialog1.Color

End Sub
```



Using Print Dialog Box



Print

General

Select Printer

Fax

Microsoft Print to PDF

Microsoft XPS Document Writer

Status: Ready

Location:

Comment:

Page Range

[✓ All]

Selection Current Page

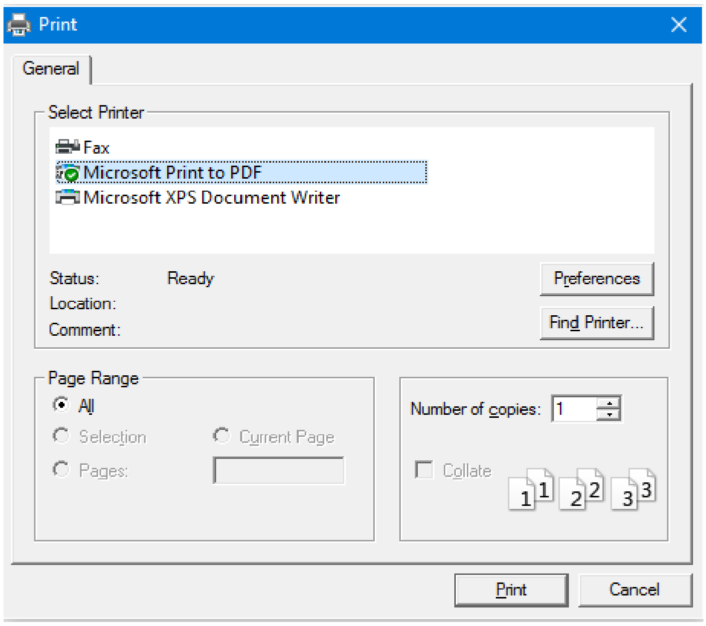
Pages:

Number of copies: 1

Collate

You show the Print dialog box with the Common Dialog Control's **ShowPrinter** method. If you know your document's length, you can set the minimum and maximum pages to print in the Common Dialog Control's Min and Max properties. Setting these properties enables the From and To page range text boxes in the Print Dialog box.

```
Private Sub Command1_Click()
  Dim c As Integer
CommonDialog1.PrinterDefault = True
CommonDialog1.Min = 0
    CommonDialog1.Max = 10
    CommonDialog1.ShowPrinter
   For c = 1 To CommonDialog1.Copies
      PrintForm
    Next c
        End Sub
```



Print

General

Select Printer

Fax

Microsoft Print to PDF

Microsoft XPS Document Writer

Status: Ready

Location:

Comment:

Page Range

[✓ All]

Selection Current Page

Pages:

Number of copies: 1

Collate

Data Access Technology

Overview of Data Access Technique

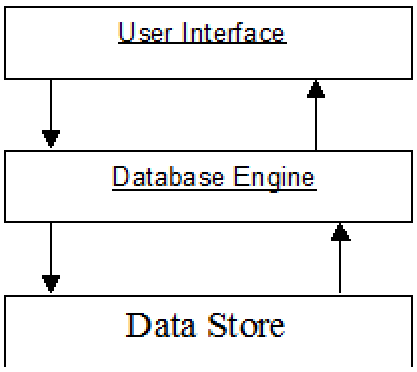
Every organization maintains data pertaining to its business, employees and clients. This data needs to be maintained such that is it easily available and can be presented in the desired format and is updated regularly.

Visual Basic provides one of the most powerful and easy front-end development environments for database management. Using Visual Basic, you can manipulate database in various formats. It also includes in-built support for Open Database Connectivity (ODBC)— an industry standard for data access from multiple database formats. Using Visual Basic, you can also build complex Client / Server database management system having remote access features.

Visual Basic Database Architecture

A Visual Basic database application consists of three components as shown in the following figure:

User Interface: This is what the user interacts with. It contains forms that display the data and enable the user to view or update it. It also includes various data access techniques to request for database services like adding or deleting records and performing queries.



Database Engine: It is contained in a set of Dynamic Link Library (DLL) files and is linked to the Visual Basic program at runtime. The engine is responsible for reading, writing and modifying the database. It also handles indexing, locking, security and referential integrity issues in the database. It contains a query processor to handle SQL queries. The database engine is logically placed between the program and the database files.

Data Store: It is the set of the files containing the database tables. For example, Microsoft Access has .mdb files containing several tables. A data store is said to be passive because it does not work on the data on its own.

Type of database application:

A database application can be categorized into three types:

Single User DBMS: In this only one user can view or manipulate data.

Multi-user DBMS: More than one user can view or manipulate data at a time. Locking and transactions are used to implement this.

Client Server DBMS: In this case, both the database engine and the data store located on a central server, multiple client applications can simultaneously request the service of the engine. The engine can access the data store and return the requested records to the client application. In Visual Basic, an application can access data using a variety of methods. These are as follows:

- Data Access Objects (DAO).
- Open Database Connectivity (ODBC).
- Remote Data Objects (RDO).
- ActiveX Data Objects (ADO).

Data Access Objects (DAO)

In Visual Basic the Microsoft Engine can be used for accessing data. The Jet engine handles the operations of storing, retrieving and updating data.

The two ways to interact with the Jet database engine are –

- Data control
- DAO

A Data control gives you limited access to database and requires minimal programming. On the other hand, DAO is a powerful object – oriented programming interface that gives you complete control over the database and its objects.

Accessing Data using Data Control

A Data Control is a standard data aware control available in Visual Basic Toolbox. It is widely used for displaying, adding and editing data from a database.

The steps to use a data control are:

Step1: Design the user interface using required controls.

Step2: Place a Data Control on the form (from the toolbox).

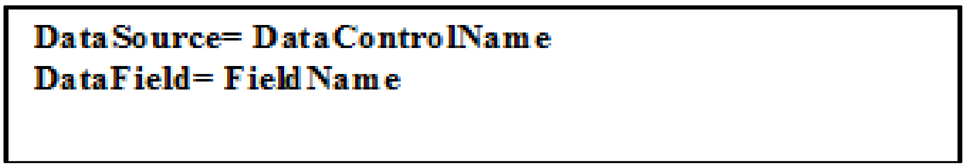
Step3: Set the Connect property of the Data Control to Ms Access.

The Connect property specifies the type of database to be opened.

Step4: Set the DatabaseName property to a specific .mdb files.

Step5: Set the RecordSource property to a Table of the specified database. The RecordSource property determines which records are included in the Recordset. The Recordset object contains a set of records.

Step6: Bind the specific Textbox with specific field of the table in the database to display that field of information to the Textbox. Controls can be bound to a field through its DataSource and DataField properties



```
DataSource= DataControlName
DataField= FieldName
```

Example:

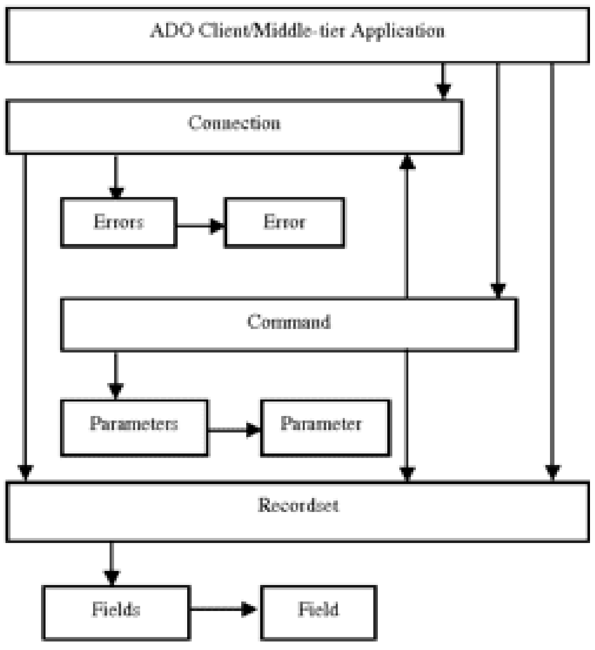
To display the name of the students to a textbox

DataSource = dtcStudent DataField = cName

Step 7 : Run the application and use the data control to view the record from the database

ADO Architecture

ActiveX Data Object (ADO) provides an object oriented programming interface for accessing data source by using an OLE provider .ADO provides you fast and easy access to all types of data. The ADO model has three main components. The Connection objects, the Command object and the Recordset object. The following illustration shows the ADO 2.1 object model -



ADO Object Model

ADO Objects

The three main components of the ADO Model are the Connection Object, Command Object and the Recordset Object. Connection Object

The Connection Object is the highest level obtained in all ADO Object Model. It is used to make connection between your application and an external data source, such Microsoft SQL Server.

Command Object

The Command Object is used to build quires including user specific parameters, in order to access records from a data source. Typically these records are returned in a Recordset Object.

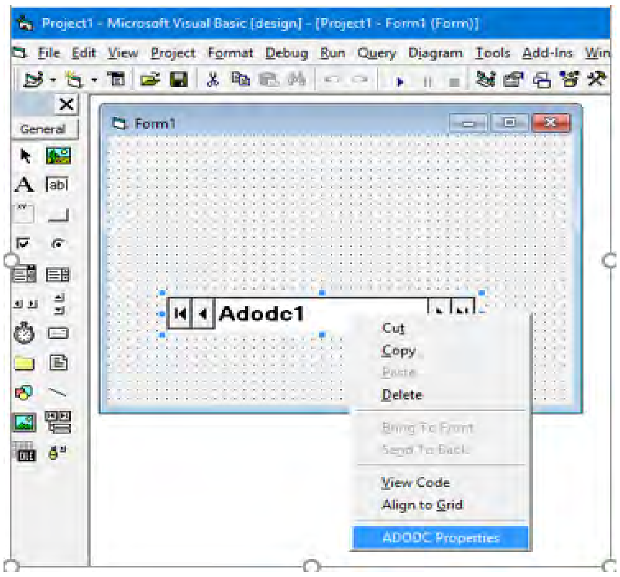
Recordset object

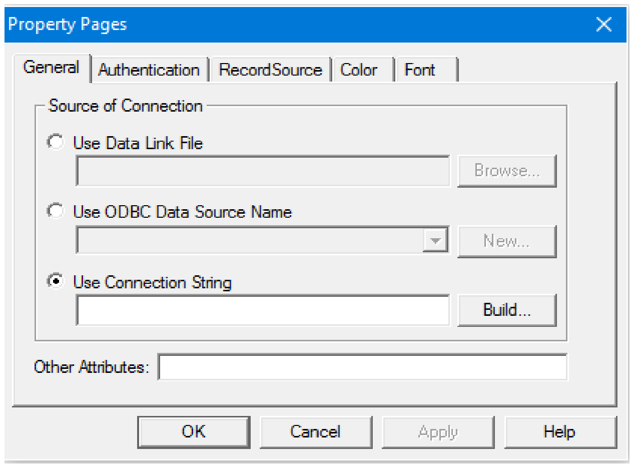
The Recordset Object is used to access the records returned from a SQL Query. Using this object, you can navigate through the records returned, modifying the existing records, add new records or delete specific record.

ADODC

The ADO Data Control (ADODC) gives developers the ease of building data bound forms that include navigation with the power and the control of using ADO. It uses Microsoft ActiveX Data Objects to quickly create connections between data bound control of using ADO to quickly create connections between data bound control and a data source.

To illustrate the connectivity using ADODC we use student database that is in the Microsoft Access. It has a table named Private _Details having the following Fields Name, Course, Batch, marks etc.


Project1 - Microsoft Visual Basic [design] - [Project1 - Form1 (Form)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Win
General
Form1
Adodc1
Cut
Copy
Paste
Delete
Bring To Front
Send To Back
View Code
Align to Grid
ADODC Properties



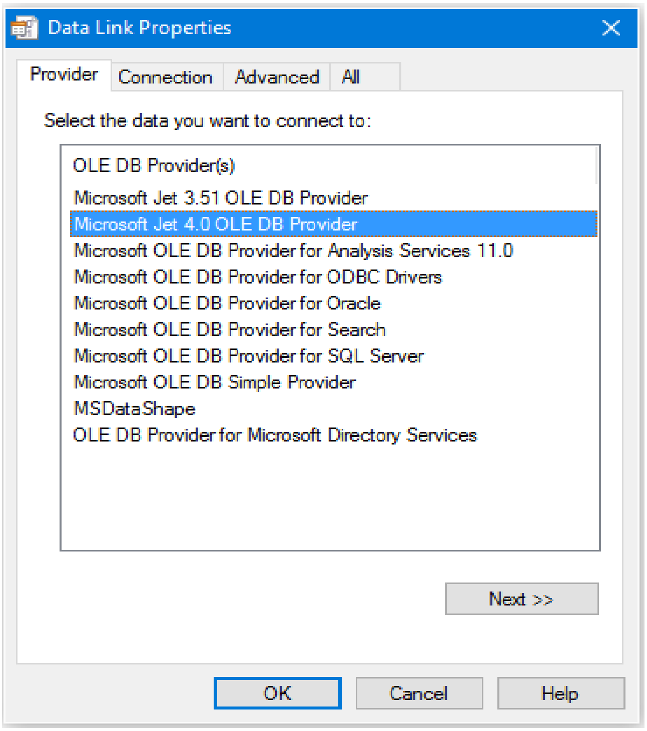
Steps to use the ADODC

Step1 : Reference the ADO Data Control i.e. add the ADODC to the toolbox. Select Components option from the Project menu. Check the Microsoft Ado Data Control 6.0 (OLEDB) option. Ok.

Step 2: Draw the ADODC from the toolbox to the Form.

Step 3: To establish the connectivity with the student database. Set the ConnectionString, CommandType and RecordSource properties of the ADO Data Control as follows-

Right Click on ADODC and select Properties option.

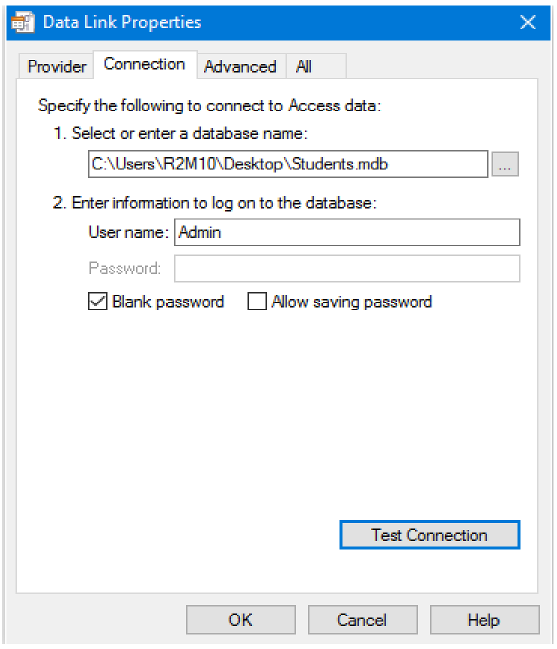

Data Link Properties

Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider for Analysis Services 11.0
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services

Since the data will be retrieved from Microsoft Access database, select Microsoft Jet 4.0 OLEDB Provider option and click on the Next>> Button.

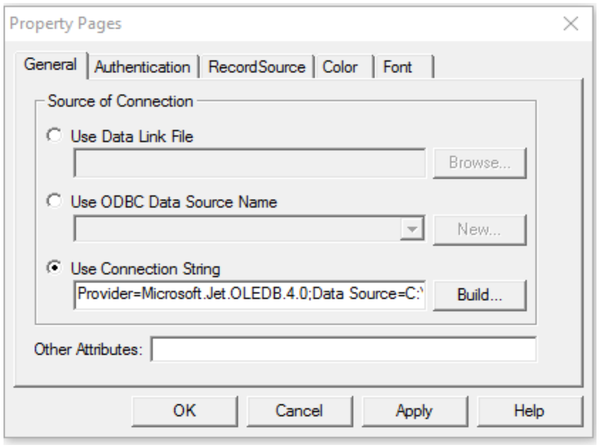
Select database name as Student


Data Link Properties

Specify the following to connect to Access data:
1. Select or enter a database name:
C:\Users\R2M10\Desktop\Students.mdb
2. Enter information to log on to the database:
User name: Admin
Password:
☑ Blank password ☐ Allow saving password

Click on the Test Connection Button, read the messages shown and click on the OK Button.

It will back to Property Page.


Property Pages

General | Authentication | RecordSource | Color | Font

Source of Connection

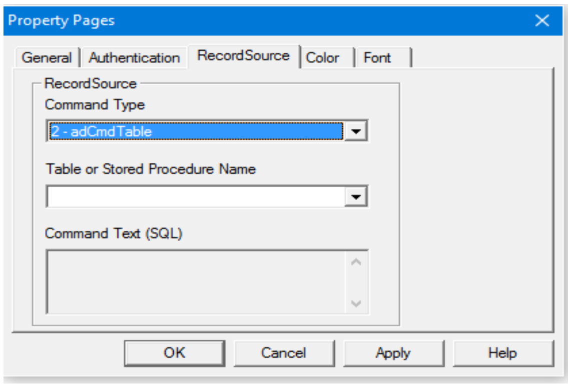
( ) Use Data Link File

( ) Use ODBC Data Source Name

(●) Use Connection String: Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\

Other Attributes:

Now select the RecordSource tab to specify the name of the table that contains the data.


Property Pages

General | Authentication | RecordSource | Color | Font

RecordSource

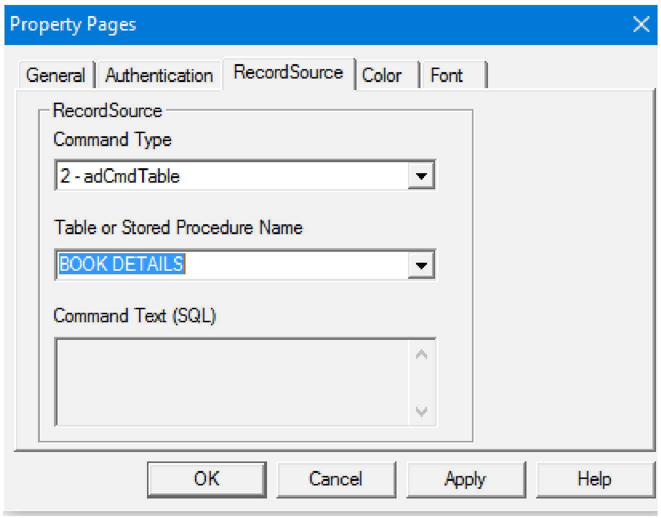
Command Type: 2 - adCmdTable

Table or Stored Procedure Name:

Command Text (SQL)

Select 2 – adcmdTable option from Command Type drop down list and select the Private_Details

option from Table or Stored Procedure Name drop down list → Click on the Ok Button


Property Pages

General | Authentication | RecordSource | Color | Font

RecordSource

Command Type: 2 - adCmdTable

Table or Stored Procedure Name: BOOK DETAILS

Command Text (SQL)

Step 4: To display specific field of information to respective textboxes, set two properties, DataSource and DataField of each textboxes.

To display Name and Course of the student to the textboxes, txtName, txtCourse

Control	Property	Value
txtName	DataSource	ADODC1
	DataField	Name
txtCourse	DataSource	ADODC1
	DataField	Course

Problem Statement:

The Excel Infocom needs to maintain the details of employee. This involves adding details of new employee, modifying and deleting details of existing employees. The details of employees that need to be maintained are :-

- Employee Id
- Dept
- Name
- Address
- Designation
- Salary

As per the problem statement the employee details needs to be maintained and stored on MS Access from there information can be displayed on Visual Basic as required for interaction.

From the above discussion it is clear that application has two parts

- Creating database in Ms access
- Creating interactive form in Visual Basic
- We are concentrating on creating the interactive part

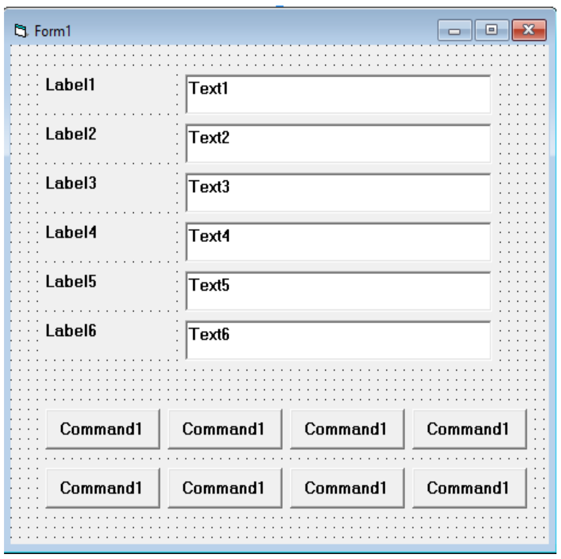
Task 1: Design the form

As Textboxes are used to accept and display the alphanumeric information, so to display the Employee Id, name, address…

We are using six textboxes and corresponding six labels for displaying the purposes of each textboxes.

As Buttons are required to start, interrupt or end a sequence of actions, therefore four command buttons are required for adding, saving, modifying and deleting records

Again four buttons required for navigation purposes that is to move to the first record, last record, next record, and previous records. So the design as follows



Controls	Properties to be changed	Changing with
Label1	Caption	Employee id
Label2	Caption	Name
Label3	Caption	Address
Label4	Caption	Department
Label5	Caption	Designation

```
Private Sub cmdFirst_Click() Adodc1.Recordset.MoveFirst

End Sub

Private Sub Form_Load()

Adodc1.Visible = False

For Each obj In Form1

If TypeOf obj Is TextBox Then

obj.Enabled = False

End If

Next

cmdsave.Enabled = False

End Sub
```

Programming with ADO

Visual Basic provides a variety of ways to accomplish any given task, and creating database applications is no exception. You saw how you could create a database application using the ActiveX Data Objects Data control

(ADODC). With the ADODC, you set up the data control, and then bound standard controls such as text boxes to it to handle displaying the data. This use of the data control and bound controls provides a means to quickly and easily create an interface for viewing and editing data.

Another approach to creating database applications is to use the ActiveX Data Objects (ADO) in code. This method requires more effort than using the ADODC, but there are some advantages to using pure code. Some of these advantages are:

You can more easily validate all the information entered by the user before it is saved to the database.

You reduce the possibility of locking conflicts in multi-user systems because your program controls when the record is locked.

You can use transaction processing to speed up data storage and to help preserve data integrity.

You can create database applications that do not require a visual interface. The data control is only good for handling programs that have a visual component.

Actually, using ADO in code and using the ADODC are not all that different. The ADODC is actually a wrapper around the ADO objects. For example, when you click one of the navigation buttons of the ADODC, you are invoking one of the Move methods of ADO. Also, when you used code to add and delete records with the ADODC, you were actually writing ADO code.

This chapter shows you how to create the various ActiveX Data Objects, how to retrieve information using the objects, and how to store new or changed information to the database.

Understanding the ActiveX Data Objects

The ActiveX Data Objects model provides Visual Basic with a robust environment for creating database applications. Each of these objects contains method and properties that control their behavior and enable them to perform certain data manipulation tasks. The ADO objects are designed to work equally well with Microsoft Jet databases and SQL databases such as SQL Server or Oracle. ADO also works with almost any Open Database Connectivity (ODBC) compatible database.

Many objects are contained within the ActiveX Data Objects. However, you will look closely at three of them in this chapter. These objects are:

Connection: This object provides the link between your programs and a data source. The data source can be a Jet database, on ODBC database, or a SQL Server data source. When you create the Connection object, you are performing the same function that the ADODC does when you set the ConnectionString property. Each Connection objects can support multiple lower-level objects, such as recordsets or commands, and other objects. The Connection object is also where transaction processing is handled in the ADO model.

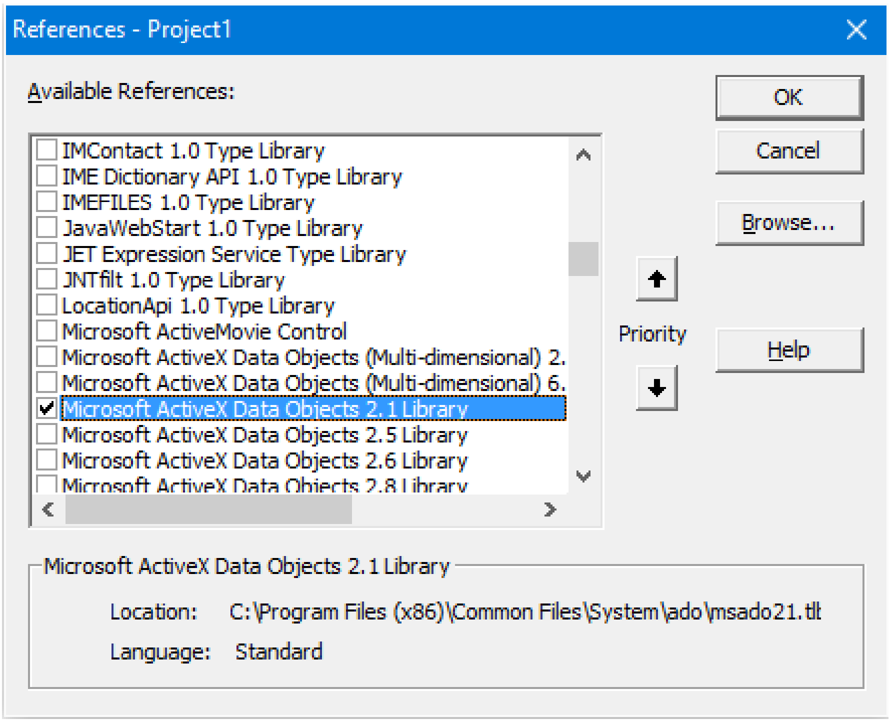
Recordset: This object is the link with the actual data in the database. It is through the Recordset object that you navigate through database records, retrieve values from fields, and update information in the database.

Command: This object is an SQL statement that can be run from your program. The Command object can contain an SQL command that was created within your program, or it can refer to a stored procedure in the database.

These three objects are the main ones used to access data in an existing database.

Creating a Connection Object

Before you can create the ADO objects, however, you have to tell Visual Basic that your program will be using ADO by setting a reference to the ADO library in the Reference dialog box. To do this, select the References option from the Project menu. The project References dialog box is displayed from which you need to select the Microsoft ActiveX Data Objects 2.1 Library as shown in the figure below. Click Ok.


References - Project1

Available References:

- [ ] IMContact 1.0 Type Library
- [ ] IME Dictionary API 1.0 Type Library
- [ ] IMEFILES 1.0 Type Library
- [ ] JavaWebStart 1.0 Type Library
- [ ] JET Expression Service Type Library
- [ ] JNTfilt 1.0 Type Library
- [ ] LocationApi 1.0 Type Library
- [ ] Microsoft ActiveMovie Control
- [ ] Microsoft ActiveX Data Objects (Multi-dimensional) 2.
- [ ] Microsoft ActiveX Data Objects (Multi-dimensional) 6.
- [✓] Microsoft ActiveX Data Objects 2.1 Library
- [ ] Microsoft ActiveX Data Objects 2.5 Library
- [ ] Microsoft ActiveX Data Objects 2.6 Library
- [ ] Microsoft ActiveX Data Objects 2.8 Library

Microsoft ActiveX Data Objects 2.1 Library

Location: C:\Program Files (x86)\Common Files\System\ado\msado21.tlb

Language: Standard

The first step to gaining access to the information in a database is to create a Connection object and establish the link to the database. As you write the code to handle creating the connections, you need to perform three steps:

- Declare a variable to hold the instance of the Connection.
- Set the properties of the Connection object.
- Use the Open method to create the link to the data source.

As you create the Connection object, you need to set two key properties – the Provider property and the ConnectionString property. The Provider property identifies the type of database with which you will be working. The property must be set to the name of an OLEDB provider. The ConnectionString property specifies the name of the database or data source that you will be accessing. This can be a Jet database or an SQL Server data source depending on your application. The following exercise shows you how to get started creating a database program and how to create the Connection object for the program. This exercise shows you how to connect to a Jet database.

\

Opening a Connection in Code

- Open a new project in Visual Basic.
- Open the References dialog box by choosing the References item from the Project menu.
- Add the reference to the Microsoft ActiveX Data Objects 2.1 Library by checking the box next to the item in the References dialog box.
- Open the Code Window for the form.
- In the Declarations section of the Code Window, declare a Connection object variable as shown in the following code:

Dim con As New Connection

In the Load event for the form, set the properties of the Connection object and open the connection using the code shown below:

With con

.ConnectionString=

”PROVIDER=MICROSOFT.JET.OLEDB.4.0;DATA SOURCE=C:\MyDocuments\STUDENT.MDB”

The ConnectionString property used in the Exercise specifies a literal string for the path and database name. If you distributing your program, you need to have a method to retrieve the user’s path for the database. You can do this by using Registry settings or by using the App.Path information if the database is in the same folder as the application.

Opening the Recordset

After you have opened a connection, the next step is to create a recordset containing the information that you want out of the data source. The recordset can contain the entire contents of a table, a few fields and records from a table, combined information from several tables, or even a single item of summary data. What is contained in the recordset depends on how you create it.

To create any recordset using the ADO, use the Open method of the Recordset object. To use this method, you need to specify the source of the records for the recordset, the name of an open Connection object, the type of a recordset to create, and the type of locking that the recordset will use. Take a look at these items one at a time.

The source of the records for the recordset can be the name of a table in the database or the name of a stored procedure (query) in the database. You can also use an SQL statement to specify the records to be used.

The Connection information for the recordset needs to be the name of a Connection object that you created in your code. This object associates the recordset with a particular database or data source. Above Exercise showed you how to create a Connection object.

The recordset type specifies how the information can be handled by your program. You can specify one of four recordset types:

Dynamic Recordset: Allows you to view the result of additions, change, and deletions made by other users. It also allows all types of movements on a recordset, such as scrolling through the records in the forward and reverse directions, and allows bookmarks (which uniquely identify a specific record), if the provider supports them.

Keyset Recordset: Behaves just like a dynamic recordset, except that it prevents you from seeing records that other users add and prevents access to records that other users delete. Data changes by other users, especially deletion and updation are visible; insertions by other users are not visible.

Static Recordset: Provides a static copy of a set of records for you to use to find data or generate reports. Always allows movement through the recordset. Additions, changes or deletions by other users are not visible.

Forward Only Recordset: Behaves identical to a static cursor except that it only allows you to scroll forward through records. This improves performance in situations where you need to make a single pass through a recordset.

The final parameter is the lock type. This setting determines how ADO locks records while the user is editing data. You can use four lock types:

- Pessimistic: Locks the record when the user starts the edit of the data in the record, usually the AddNew or Edit methods of the recordset.
- Optimistic: Locks the record only when the changes to the data are saved to the database using the Update method.
- Read-Only: Specifies that the data in the recordset cannot be modified. BatchOptimistic: Handles batch updates of multiple records in the recordset.

Summary:

Visual Basic 6.0 was a popular, user-friendly tool for building Windows applications. Its ease of use, integration with databases, and support for event-driven programming made it ideal for developers building business applications and small-scale software. Despite being largely replaced by VB.NET and other modern tools, VB 6.0 had a lasting impact on Windows application development.

Check your Understanding

Question 1

How many default controls are available in VB?

a) 20
b) 21
c) 23
d) 22

Question 2

Visual Basic maintains a project file with the extension ________

a) .frm
b) ,vbp
c) .vbs
d) .cls

Question 3

The form module has file extension ________

a) ,frb
b) .fra
c) .frm
d) .fru

Question 4

When the user clicks a button, _________ is triggered.

a) An event
b) A method
c) A setting
d) A property

Question 5

“Form_Load” – the “Load” is the name of

a) Keyword
b) Procedure
c) Syntax
d) All of the above

Question 6

Visual Basic is

a) Operating System
b) Programming Language
c) Application Software
d) All of the above

Question 7

AutoSize property is for

a) Label
b) Combo Box
c) Button
d) All of the above

Question 8

Dim a as Integer – Where Integer is

a) String
b) Number
c) Boolean
d) All of the above

Question 9

VBCritical is used for

a) MsgBox
b) TextBox
c) InputBox
d) ComboBox

Question 10

Key field of the ConnectionString property are

a) Mode
b) Provider and Data Source
c) Provider
d) Data Source

Chapter 3: Overview of Visual Basic .NET

Introduction

Visual Basic .NET (VB.NET) is an object-oriented programming language developed by Microsoft. It is part of the .NET framework and is a developed version of the older Visual Basic (VB) language. It is designed to be easy to learn and use, with very simple syntax, clear, and readable, making it a popular choice for beginners and experienced too.

Key Features VB.Net

1) Object-Oriented Programming (OOP): VB.NET supports all the features of object-oriented programming, including classes, objects, inheritance, polymorphism, and encapsulation. This makes it powerful for building scalable and maintainable applications.
2) Integration with .NET Framework: VB.NET is tightly integrated with the .NET framework, which provides a vast library of pre-built classes and components for various types of applications, including web, desktop, and mobile applications. It uses the Common Language Runtime (CLR), which allows for the execution of code across different languages.
3) Rich Library Support: The .NET Framework includes libraries for everything from graphical user interfaces (GUIs) to data access and web services. VB.NET developers can leverage these libraries to create feature-rich applications.
4) Event-Driven Programming: VB.NET is event-driven, which means it responds to user actions such as button clicks, mouse movements, and keyboard presses. This makes it ideal for developing desktop and web applications with dynamic user interfaces.
5) Cross-Language Compatibility: Since VB.NET is part of the .NET family, it can interoperate with other .NET languages like C# and F#. This cross-language compatibility allows developers to integrate code written in different languages seamlessly.
6) Modern Language Features: VB.NET includes modern programming features such as:
   a) Garbage Collection: Automatic memory management to free up unused resources.
   b) Exception Handling: Structured error handling with Try, Catch, Finally blocks.
   c) LINQ (Language Integrated Query): Makes it easier to query data in collections, databases, and XML documents using a SQL-like syntax.
   d) Asynchronous Programming: Features such as Async and Await to simplify writing asynchronous code for scalable applications.
7) Web and Desktop Development: VB.NET can be used for both desktop and web applications:
   a) Windows Forms (WinForms): A framework for creating traditional desktop applications with rich user interfaces.
   b) ASP.NET: A framework for building dynamic web applications and services. VB.NET can be used to write server-side code for web applications.

Uses of VB.NET

1) Web Applications: Creating dynamic websites and web services using ASP.NET.
2) Windows Applications: Building desktop software for Windows using WinForms or WPF (Windows Presentation Foundation).
3) Mobile Applications: Though not as commonly used for mobile development today, VB.NET can be used with Xamarin for cross-platform mobile app development.
4) Database Applications: Developing data-driven applications, connecting to databases like SQL Server, and handling data with ADO.NET or Entity Framework.
5) Enterprise Applications: VB.NET is used for large-scale enterprise solutions, particularly in organizations that rely heavily on Microsoft technologies.

Example of Simple VB.NET Code:

```
Module Module1

Sub Main()

Dim name As String

Console.WriteLine("Enter your name:")

name = Console.ReadLine()

Console.WriteLine("Hello, " & name)

Console.ReadKey()

 End Sub
```

This simple VB.NET program asks for the user's name and then prints a greeting. It shows the language's straightforward syntax and ease of use.

Summary:

VB.NET is a versatile and powerful language suitable for developing a wide range of applications on the Microsoft .NET platform. It is easy to learn, integrates well with other .NET languages, and provides strong support for both desktop and web applications. While its popularity may have waned in favour of other languages like C#, it still remains a valuable tool for developers working within the .NET ecosystem.

Check your understanding

Question 1

Which of the following is a key feature of VB.NET?

a) It is an interpreted language.
b) It supports Object-Oriented Programming.
c) It is not compatible with the .NET Framework.
d) It can only be used for Windows desktop applications.

Question 2

What does the "CLR" in VB.NET stand for?

a) Common Language Runtime
b) Code Level Runtime
c) Common Library Repository
d) Code Level Repository

Question 3

Which of the following libraries is commonly used in VB.NET for web development?

a) ASP.NET
b) WPF
c) Windows Forms
d) ADO.NET

Question 4

What does LINQ stand for in VB.NET?

a) Local Internet Query
b) Language Integrated Query
c) Library Integrated Query
d) Local Interface Query

Question 5

Which of the following is the default IDE for developing VB.NET applications?

a) Sublime Text
b) IntelliJ IDEA
c) Visual Studio
d) Eclipse

Chapter 4: RDBMS concepts with MS-Access

Introduction

Microsoft Access 2016 is a database management system (DBMS) that combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools. It enables users to create, manage, and manipulate databases efficiently. MS Access 2016 is part of the Microsoft Office Suite and is widely used for small-to-medium-sized database applications.

Features of MS Access 2016

User-Friendly Interface

- Provides a simple and intuitive interface for creating and managing databases.
- Ribbon menu with categorized tabs for easy navigation.

Templates

- Built-in templates for common database types like contacts, tasks, inventory, etc.

Relational Database Management

- Links multiple tables using relationships for efficient data organization and reduced redundancy.

Queries

- Build queries to retrieve, update, or delete data with ease.
- Supports SQL (Structured Query Language) for advanced users.

Forms and Reports

- Create user-friendly input forms for data entry.
- Generate reports to summarize and present data effectively.

Macros and VBA (Visual Basic for Applications)

- Automate repetitive tasks using macros and VBA for custom scripting .

Data Import and Export

- Import data from Excel, SharePoint, or SQL Server and Export Data

Multi-User Support

- Allows multiple users to work on the database simultaneously when stored on a shared network.

Integration with Microsoft Office Applications

- Seamless integration with Excel, Word, and Outlook for extended functionality.

Security Features

- Password protection for databases.
- User-level permissions for access control.

Core Components of MS Access 2016

Tables

- Store data in rows and columns.
- Define fields with specific data types (e.g., text, number, date).

Forms

- User-friendly interfaces for data entry and editing.
- Customizable design with buttons, dropdowns, and more.

Queries

- Retrieve data based on specific criteria.
- Use the Query Wizard for simplified query creation.

Reports

- Design reports to display data in a structured and readable format.
- Include calculations, summaries, and charts.

Macros

- Automate tasks like opening forms, running queries, or validating data.

Modules

- VBA code for advanced programming and custom features.

Getting Started with MS Access 2016

1. Launch Access 2016

   a. Open Access from the Microsoft Office Suite.

2. Create a Database

   a. Use a blank database or select a template from the start screen.

3. Build Tables

   a. Define fields and set data types.

   b. Add primary keys for unique identification of records.

4. Establish Relationships

   a. Link tables using primary and foreign keys.

5. Input Data
    a. Use datasheet view or forms for data entry.
6. Create Queries
    a. Use the Query Wizard or SQL view to define and run queries.
7. Design Forms and Reports
    a. Use drag-and-drop tools in the form and report design view.
8. Save and Secure
    a. Save the database regularly.
    b. Apply passwords or user permissions for added security.

Advantages of MS Access 2016

1. Ease of Use: Suitable for beginners and advanced users.
2. Cost-Effective: Affordable compared to larger DBMS like SQL Server.
3. Rapid Development: Templates and wizards simplify database creation.
4. Integration: Excellent compatibility with other Microsoft tools.
5. Scalable: Suitable for small-to-medium-sized databases.

Limitations of MS Access 2016

- Limited Scalability: Not ideal for large-scale or enterprise-level databases.
- Performance Issues: Slower with large data volumes or multiple concurrent users.
- Platform Dependency: Only runs on Windows systems.
- Security Concerns: Basic security measures might not be sufficient for sensitive data.

Practical Applications

1. Small Business Management
    - Track inventory, manage customer databases, and handle billing.
2. Education
    - Store and manage student records, course information, and grades.
3. Healthcare
    - Maintain patient records and appointment schedules.
4. Non-Profits
    - Manage donor information and fundraising activities.

The Navigation Pane in MS Access

The Navigation Pane in Microsoft Access is a central feature that helps you manage and access the objects in your database. It serves as an organized directory where all the components of your database, such as tables, queries, forms, reports, macros, and modules, are displayed.

Purpose of the Navigation Pane

Access Database Objects

- Provides an overview of all objects in the database for easy selection and usage.

Organize Objects

- Allows you to group and categorize database objects for better organization and faster navigation.

Perform Actions

- Enables opening, renaming, deleting, or managing properties of database objects.

Default Layout of the Navigation Pane

By default, the Navigation Pane organizes objects under broad categories like:

- Tables
- Queries
- Forms
- Reports
- Macros
- Modules

You can also customize the grouping based on your preference.

Key Features of the Navigation Pane

1. Search Bar
    - Quickly find objects by typing their name in the search box.
2. Filter by Object Type
    - Display objects based on type, such as Tables, Queries, or Forms.
3. Custom Groups
    - Create and organize objects into custom groups for specific workflows.
4. Show or Hide
    - Use the F11 key to toggle the Navigation Pane on or off for more workspace.

Customizing the Navigation Pane

1. Collapse/Expand Groups
    - Click on the arrow next to a category name to collapse or expand its contents.

2. Change Display Options
    - Right-click on the Navigation Pane and select Navigation Options to customize how objects are displayed.
3. Create Custom Categories
    - Use the Navigation Options menu to define custom categories and groups.
4. Sort Objects
    - Sort objects alphabetically or by creation date for better visibility.

Managing Objects with the Navigation Pane

- Open an Object: Double-click the object name.
- Rename an Object: Right-click on the object and choose Rename.
- Delete an Object: Right-click and select Delete (use with caution as it permanently removes the object).
- Export an Object: Right-click and choose Export to transfer data to another format.

Tips for Using the Navigation Pane

- Use Ctrl + F to quickly locate an object.
- Create shortcuts for frequently used objects to streamline workflow.
- Organize complex databases with custom groups for faster access.

Creating a Table in MS Access

A table in MS Access is a fundamental object where data is stored in rows and columns. Each table contains fields (columns) that represent data types and records (rows) that store the actual data.

Steps to Create a Table

Using Datasheet View

The Datasheet View provides a spreadsheet-like interface for creating and managing tables.

1. Open MS Access and either create a new database or open an existing one.
2. Click on the Create tab in the ribbon.
3. Select Table to create a new table in Datasheet View.
4. Start adding fields by clicking on the Add Field column.
    a. Enter a field name, then choose the Data Type from the dropdown.
    b. For example: \`Name\` (Short Text), \`Age\` (Number), \`Date of Birth\` (Date/Time).
5. Input data directly into the rows for records.
6. Save the table:
    c. Click File > Save or press Ctrl + S.
    d. Provide a name for the table and click OK.

Using Design View

The Design View allows you to define the structure of the table, including field properties and primary keys.

1. Open MS Access and go to the Create tab.
2. Select Table Design to create a table in Design View.
3. Define the fields:
   a. Enter the Field Name in the first column.
   b. Choose a Data Type for each field.
   c. Optionally, provide a description for better documentation.
4. Set the Primary Key:
   d. Right-click the field that should uniquely identify each record (e.g., `ID`).
   e. Choose Primary Key.
5. Save the table:
   f. Click File > Save or press Ctrl + S.
   g. Provide a name for the table and click OK.

Field Data Types

Common data types in MS Access include:

- Short Text: For small text entries (e.g., names, addresses).
- Long Text: For lengthy text entries (e.g., descriptions).
- Number: For numerical data.
- Date/Time: For dates and times.
- Currency: For financial values.
- Yes/No: For Boolean data (e.g., true/false).
- AutoNumber: Automatically generates unique values.

Setting Table Properties

1. Field Size: Limits the number of characters or digits.
2. Default Value: Specifies a value automatically assigned to a field.
3. Validation Rule: Defines conditions for valid data entry.
4. Input Mask: Formats data entry (e.g., phone numbers).
5. Required: Ensures a field cannot be left blank.

Best Practices for Creating Tables

1. Plan Your Table Design: Clearly identify the data you need to store and the relationships between tables.
2. Set a Primary Key: Ensure every table has a unique identifier.
3. Use Appropriate Data Types: Choose data types that match the kind of data you will store.
4. Avoid Data Duplication: Normalize your database to minimize redundancy.

Example Table

ID (Primary Key)	Name	Age	DOB	Email
1	J Das	28	1995-06-15	jdoe@yctc.com
2	S Sen	34	1989-02-23	smith@yctc.com

Primary Key in MS Access

A Primary Key is a field or a combination of fields in a table that uniquely identifies each record. It ensures that no duplicate values exist in the key field(s), and every record can be uniquely distinguished.

Key Characteristics of a Primary Key

1. Uniqueness: Each record in the table must have a unique value in the primary key field.
2. No Null Values: A primary key cannot contain `NULL` values.
3. Single or Composite: A primary key can be a single field or a combination of multiple fields (composite key).

Purpose of a Primary Key

1. Uniquely Identifying Records: Prevents duplicate records and allows for quick and accurate record retrieval.
2. Data Integrity: Ensures the accuracy and consistency of data in the database.
3. Establishing Relationships: Helps link tables in relational databases through foreign keys.

How to Set a Primary Key in MS Access

In Design View

1. Open the table in Design View.
2. Click on the field that you want to set as the primary key.
3. Click the Primary Key button in the toolbar (under the Design tab).
   a. A key icon will appear next to the field name, indicating it's the primary key.
4. Save the table.

In Datasheet View

1. Open the table in Datasheet View.
2. Right-click on the column header of the field you want to set as the primary key.
3. Select Primary Key from the context menu.
4. Save the table.

For Composite Keys

1. Open the table in Design View.
2. Hold Ctrl and click the row selectors (grey boxes) for the fields you want to combine as the primary key.
3. Click the Primary Key button in the toolbar.
4. Save the table.

Best Practices for Choosing a Primary Key

1. Use AutoNumber for tables that require system-generated unique values.
2. Use meaningful fields when possible, such as `EmployeeID` or `OrderID`.
3. Avoid using large or frequently updated fields as primary keys to maintain performance.
4. Ensure the field has no chance of duplication.

StudentID (Primary Key)	Name	Age	Course
1	Partha Das	20	Computer Science
2	Debjit Dutta	22	Business Admin
3	Nilanjan Chakraborty	18	Engineering

Relationship in MS Access

A relationship in MS Access defines how data in two or more tables is connected. By establishing relationships, you can ensure data integrity, reduce redundancy, and create a unified database structure.

Types of Relationships

1. One-to-One (1:1):
    - Each record in Table A corresponds to one record in Table B.
    - Example: A table for employees and a table for employee credentials.
2. One-to-Many (1:N):
    - A single record in Table A relates to multiple records in Table B.
    - Example: A customer can have multiple orders, but each order belongs to one customer.
3. Many-to-Many (M:N):
    - Records in Table A relate to multiple records in Table B and vice versa.
    - Example: Students enrolled in multiple courses, and courses taken by multiple students.
    - Requires an intermediate junction table to implement.

Creating Relationships

1. Establishing Relationships

1. Open your MS Access database.
2. Go to the Database Tools tab and click on Relationships.
3. In the Relationships window:
   i. Click Show Table to add the tables you want to connect.
   ii. Drag a field (usually the primary key) from one table to a related field (foreign key) in another table.
4. The Edit Relationships dialog box will appear. Configure as needed:
   i. Select the relationship type (e.g., One-to-Many).
   ii. Check Enforce Referential Integrity to maintain data accuracy.
5. Click Create to establish the relationship.

2. Referential Integrity

a) Ensures that relationships between tables remain consistent.
b) Example: If a customer record is deleted, Access prevents deletion if that customer has related orders unless cascading is enabled.

Options:

- Cascade Update: Updates related fields when a primary key value changes.
- Cascade Delete: Deletes related records when a primary key value is deleted.

Viewing Relationships

a. Use the Relationships Window to visualize how tables are connected.
b. Relationships are represented with lines:
   i. 1 indicates one side of the relationship.
   ii. ∞ (infinity symbol) indicates many sides.

Using Relationships

1. Queries: Fetch data from multiple tables based on their relationships.
2. Forms and Reports: Combine data dynamically using related tables.
3. Data Integrity: Prevent orphaned records (e.g., orders without a valid customer).

Benefits of Relationships

- Reduces Redundancy: Avoids duplication of data.
- Ensures Data Accuracy: Enforces rules for data consistency.
- Simplifies Queries: Makes it easier to retrieve related data.

Relationships are the backbone of relational databases like MS Access, enabling efficient and organized data management.

Creating Queries

Creating Queries in MS Access

A query in MS Access allows you to retrieve, filter, and analyze data from tables. It's a powerful way to extract specific information and perform calculations or updates on your data.

Types of Queries in MS Access

1. Select Query: Retrieves specific data from one or more tables.
2. Action Query: Performs actions like adding, updating, or deleting data.
   a) Append Query
   b) Update Query
   c) Delete Query
   d) Make-Table Query
3. Parameter Query: Prompts the user for input to filter the data.
4. Crosstab Query: Summarizes data in a matrix format.
5. SQL Query: Directly writes SQL statements for advanced customization.

Steps to Create a Query

Using Query Wizard

1. Open your MS Access database.
2. Go to the Create tab in the ribbon.
3. Click on Query Wizard in the Queries group.
4. Choose the type of query to create (e.g., Simple Query Wizard).
5. Select the table(s) and field(s) to include in the query.
6. Specify any sorting order or summary options if needed.
7. Click Finish to generate the query.

Using Query Design

1. Go to the Create tab and click on Query Design.
2. Add the table(s) or query(s) you want to use in the query.
3. Drag the fields you need into the grid at the bottom.
4. Set criteria for filtering data (e.g., `>100` for numbers greater than 100).
5. Run the query by clicking Run (red exclamation mark icon) in the ribbon.
6. Save the query by clicking Ctrl + S and give it a descriptive name.

Writing SQL Queries

1. Go to the Create tab and click on Query Design.
2. Switch to SQL View by selecting it from the View drop-down menu.
3. Write your SQL statement (e.g., `SELECT * FROM Employees WHERE Salary > 50000;`).
4. Run the query and save it as needed.

Adding Criteria to a Query

- Use the Criteria row in Query Design to filter data.
- Examples:
  - Numeric Values: `>5000`, `<1000`
  - Text Matching: `"Marketing"` (exact match), `Like "Mar*"` (partial match)
  - Date Range: `Between #01/01/2023# AND #12/31/2023#`

Examples of Common Queries

1. Simple Select Query

- Retrieve all employees with a salary above $50,000.

```
SELECT EmployeeID, Name, Salary
FROM Employees
WHERE Salary > 50000;
```

2. Parameter Query

- Prompts user to input a department name.

```
SELECT *
FROM Employees
WHERE Department = [Enter Department Name];
```

3. Update Query

- Increase salaries by 10% in the Sales department.

```
UPDATE Employees
SET Salary = Salary * 1.1
WHERE Department = 'Sales';
```

4. Crosstab Query

- Show total sales by region and quarter.

Best Practices for Queries

1. Use Indexes: For faster retrieval of data.
2. Minimize Fields: Only include necessary fields to reduce query load.
3. Use Aliases: Simplify complex field names for better readability.
4. Test Criteria: Ensure your filters provide accurate results.

Queries are the backbone of data retrieval and management in MS Access, offering flexibility and efficiency for analyzing and manipulating data.

Creating Report

Creating Reports in MS Access

A report in MS Access is used to present data in a structured and visually appealing format. Reports are often used for printing or sharing summaries, making them useful for analysis and decision-making.

Steps to Create a Report

Using Report Tool

1. Open your database and ensure your table or query contains the required data.
2. Go to the Create tab in the ribbon.
3. Click on Report in the Reports group.
4. MS Access automatically generates a simple report based on the selected table or query.
5. Adjust the layout in Layout View or Design View as needed.
6. Save the report using Ctrl + S and name it appropriately.

Using Report Wizard

1. Go to the Create tab and click on Report Wizard.
2. Select the table or query that contains the data for the report.
3. Choose the fields to include in the report.
   a. Use the > button to select specific fields or >> to select all fields.
4. Specify grouping levels if needed (e.g., group by department or region).
5. Choose sorting options for the data (e.g., sort by name or date).
6. Select a layout style (e.g., Columnar, Tabular, or Justified).
7. Choose a report title and click Finish to generate the report.

Using Blank Report

1. Go to the Create tab and click on Blank Report.
2. Use the Field List Pane to drag and drop fields from a table or query onto the report.
3. Design the layout manually using the tools in the Design or Format tabs.
4. Save the report after customization.

Designing and Customizing the Report

Report Views:

1. Report View: View data interactively but cannot make design changes.
2. Print Preview: See how the report will look when printed.
3. Layout View: Make quick adjustments while viewing live data.
4. Design View: Fine-tune the layout, controls, and formatting.

Adding Controls:

1. Use Text Boxes, Labels, Images, and Charts to enhance the report.
2. Add a Title or a Logo using the tools in the ribbon.

- Formatting:

1. Adjust fonts, colors, and alignments for better readability.
2. Use borders, shading, and conditional formatting to highlight specific data.

Examples of Reports

1. Sales Summary Report:
   a. Fields: Date, Sales Amount, Region.
   b. Grouped by Region and sorted by Date.
   c. Includes totals and averages for each region.
2. Employee Performance Report:
   a. Fields: Employee Name, Department, Performance Score.
   b. Grouped by Department with scores displayed in a bar chart.
3. Inventory Report:
   a. Fields: Product Name, Stock Level, Reorder Level.
   b. Highlights low stock items using conditional formatting.

Benefits of Reports in MS Access

1. Data Presentation: Summarizes complex data into easy-to-read formats.
2. Customization: Flexible layouts to suit specific requirements.
3. Print-Ready: Ideal for generating professional reports for meetings or sharing.

Reports are an essential feature in MS Access, transforming raw data into actionable insights with professional-quality output.

Creating a Form

A form in MS Access is a user-friendly interface for entering, viewing, and editing data in tables. It simplifies data interaction, especially for non-technical users.

Steps to Create a Form

Using Form Tool

1. Open MS Access and ensure your table is ready.
2. Go to the Create tab in the ribbon.
3. Click on Form in the Forms group.
4. MS Access automatically generates a form based on the selected table.
5. Save the form by clicking File > Save or pressing Ctrl + S, and give it a name.

Using Form Wizard

1. Go to the Create tab.
2. Click on Form Wizard in the Forms group.
3. Select a table or query for your form and choose the fields to include.
   a. Use the > button to select specific fields or >> to select all fields.
4. Click Next and select a layout style (e.g., Columnar, Tabular, Datasheet).
5. Choose a form name and click Finish.
6. The form opens in Form View for data entry.

Using Blank Form

1. Go to the Create tab and click on Blank Form.
2. Use the Field List Pane on the right to drag and drop fields from a table or query onto the form.
3. Design the form manually using tools in the Design or Format tabs.
4. Save the form after customization.

Designing a Form

Use the Design View to adjust the layout, add controls, or format elements:

1. Right-click on the form and select Design View.
2. Add labels, buttons, text boxes, and other controls from the Design tab.
3. Arrange fields and customize colors, fonts, and sizes.

- Add Combo Boxes or List Boxes to make data entry easier for fields with predefined options..

Advantages of Using Forms

1. User-Friendly: Simplifies data entry and navigation.
2. Customizable Interface: Tailor the form to suit user needs.
3. Data Validation: Enforce rules during data entry.
4. Enhanced Functionality: Use buttons and macros for advanced features like navigation, printing, or filtering records.

Summary:

Microsoft Access is a powerful yet user-friendly tool for creating, managing, and analyzing databases. It combines the features of a relational database system with an intuitive interface, making it suitable for both novice users and more advanced developers. It is commonly used in small businesses, workgroups, and departments where users need to organize and process data efficiently without needing complex database administration skills. However, for very large or enterprise-level applications, more robust database systems like SQL Server may be necessary.

Check your Understanding

Question 1

We can’t create index or primary key on more than ________ fields

a) 10
b) 16
c) None of the above
d) 1

Question 2

Which type of field is incremented automatically?

a) AutoNumber
b) Auto Value
c) Auto Increment
d) Auto size

Question 3

Which of the following type of queries are action queries?

a) Update queries
b) Crosstab Queries
c) Append Queries
d) Both 1 and 3

Question 4

It is a query that when run displays its own dialog box prompting you for information, such as

a) Select
b) crosstab
c) Parameter
d) Append

Question 5

Which of the following is not a valid data type in MS-Access

a) Memo
b) Picture
c) Currency
d) Auto number

Question 6

A subset of characters within a data field is known as

a) Byte
b) File
c) Record
d) Data string

Question 7

This type of query summarizes large amounts of data in easy-to-read, row and column format

a) Append
b) Rows
c) Select
d) Crosstab

Question 8

This Key Uniquely Identifies Each Record

a) Field Name
b) Unique Key
c) Primary Key
d) Key Record

Question 9

Which Of the Field Has Width 8 Bytes?

a) Memo
b) number
c) Date/Time
d) Hyperlink

Question 10

To print information from a table, the __________ tool is the best choice.

a) form
b) report
c) Query
d) macro

Chapter 5: Internet and Email

Introduction:

The internet is a vast global network that connects millions of computers, devices, and systems, allowing them to communicate and share data with each other. It has become an essential part of modern life, enabling access to a wide range of services, from communication and entertainment to education and commerce.

Key Aspects of the Internet

1. Structure:
   - i. The internet is a decentralized network made up of interconnected computers, servers, and routers.
   - ii. It uses a standard set of protocols called TCP/IP (Transmission Control Protocol/Internet Protocol) to manage the transmission of data.
   - iii. Key components include the World Wide Web (WWW), email, social media platforms, cloud services, and much more.
2. History:
   - iv. The internet began as ARPANET in the late 1960s, a project funded by the U.S. Department of Defense. Its original purpose was to create a communication network that could withstand disruptions in case of war.
   - v. The introduction of the World Wide Web by Tim Berners-Lee in the 1990s revolutionized how people interact with the internet, allowing for the creation of websites and easy access to information.
3. Internet Services:
   - vi. Web browsing: The most common use of the internet, where users access websites using browsers like Chrome, Firefox, or Safari.
   - vii. Email: Electronic communication via platforms like Gmail, Yahoo Mail, or Outlook.
   - viii. Social Media: Platforms like Facebook, Twitter, Instagram, and TikTok allow users to connect and share content.
   - ix. E-commerce: Websites like Amazon, eBay, and Alibaba have transformed the way people shop online.
   - x. Streaming: Services like Netflix, Spotify, and YouTube offer media streaming, from movies to music and videos.
   - xi. Cloud Computing: Services like Google Drive, Dropbox, and iCloud provide online storage and computing resources.
4. Cybersecurity:
   - xii. As the internet has grown, so has the importance of securing personal and business data from threats like hacking, phishing, viruses, and malware.
   - xiii. Methods such as encryption, firewalls, and secure browsing practices are essential in maintaining privacy and security online.

5. Internet Governance and Access:
   - xiv. The internet is governed by various organizations like ICANN (Internet Corporation for Assigned Names and Numbers) and IETF (Internet Engineering Task Force), which help manage domain names and technical standards.
   - xv. Global access to the internet is uneven, with disparities in infrastructure and availability in different regions, especially between developed and developing countries.
6. Impact on Society:
   - xvi. Communication: The internet has revolutionized communication through instant messaging, video calls, and social media, shrinking distances and enabling global connections.
   - xvii. Education: It has transformed learning with online courses, research tools, and educational platforms like Coursera, Khan Academy, and edX.
   - xviii. Work and Business: The rise of remote work, online businesses, and digital marketing has reshaped industries.
   - xix. Social Changes: The internet has had a profound impact on social dynamics, including changes in how people form relationships, express their opinions, and share their lives with others.
7. Challenges:
   - xx. Digital Divide: Not everyone has equal access to the internet, which creates gaps in opportunities for education, work, and social participation.
   - xxi. Privacy Concerns: With the collection of vast amounts of personal data, the internet raises concerns about how information is used, stored, and shared.
   - xxii. Misinformation and Fake News: The rapid spread of misinformation on the internet, especially on social media platforms, has become a significant issue in shaping public opinion and political discourse.

Overview of Email

Email (electronic mail) is one of the most widely used forms of communication in the modern world. It allows individuals and organizations to send, receive, and store messages electronically over the internet. Email has become an essential tool for personal, professional, and business communication, offering a fast, efficient, and cost-effective way to share information.

Key Aspects of Email

1. How Email Works:
   - Email Address: An email address identifies the sender and recipient of the message. It typically follows the format username@domain.com, where "username" represents the individual's or organization's identifier, and "domain.com" is the email service provider's domain (e.g., Gmail, Yahoo).
   - Mail Servers: Emails are sent and received through mail servers, which use protocols such as SMTP (Simple Mail Transfer Protocol) for sending messages and IMAP (Internet Message Access Protocol) or POP3 (Post Office Protocol) for receiving and retrieving them from mail servers.
   - Inbox, Sent, and Other Folders: Email services typically organize messages into folders such as Inbox, Sent, Drafts, and Trash. Users can manage and sort their emails to keep their communication organized.
2. Components of an Email:
   - Subject Line: A brief summary of the email's content, helping the recipient understand the purpose of the email.
   - Recipient(s): The primary recipient (To), secondary recipients (CC, which stands for Carbon Copy), and hidden recipients (BCC, for Blind Carbon Copy).
   - Body: The main content of the email, which can be plain text, rich text (formatted), or HTML (for emails with images and styling).
   - Attachments: Files (documents, images, etc.) that are sent along with the email.
   - Signature: A closing section that often includes the sender's name, title, company, or contact information.
3. Email Protocols:
   - SMTP: Used to send emails from the sender to the mail server and from the server to the recipient's mail server.
   - IMAP: Allows the recipient to access and manage their emails on the mail server without downloading them (ideal for accessing emails across multiple devices).
   - POP3: Downloads emails from the server to the recipient's device and usually removes them from the server after they are downloaded, making it less flexible than IMAP.
4. Email Clients and Services:
   - Email Clients: Programs or applications used to access and manage emails, such as Microsoft Outlook, Apple Mail, or Mozilla Thunderbird.
   - Webmail: Services that allow users to access their email via a web browser, such as Gmail, Yahoo Mail, and Outlook.com. These services are convenient as they don’t require specific software and can be accessed from any internet-enabled device.

5. Email Etiquette:
   - Professionalism: Email is often used in professional and business settings, so it is important to maintain proper etiquette, including clear language, a professional tone, and respectful communication.
   - Clarity: Emails should be concise and to the point, avoiding unnecessary jargon or ambiguity.
   - Subject Line: A clear and relevant subject line is essential for effective communication, especially in a professional setting.
   - Replying: When replying to emails, it's important to reply to the right recipients (use "Reply All" sparingly) and acknowledge key points from the original message.
6. Security and Privacy:
   - Spam: Unsolicited or unwanted emails, often used for advertising or malicious purposes. Spam filters help users manage and block unwanted messages.
   - Phishing: A type of fraud where scammers impersonate trusted entities to steal sensitive information like passwords, credit card numbers, or bank account details.
   - Encryption: For privacy, some email services offer encryption, ensuring that the email content is only readable by the intended recipient.
   - Two-Factor Authentication (2FA): Enhances security by requiring both a password and a secondary form of verification (such as a code sent to your phone) when logging into an email account.
8. Advantages of Email:
   - Speed: Emails can be sent and received almost instantaneously, making it one of the fastest ways to communicate.
   - Global Reach: Email allows users to communicate with people worldwide at any time of day or night.
   - Documentation: Emails provide a written record of communication, which can be referenced later for clarity or legal purposes.
   - Cost-Effective: Email eliminates the need for physical postage or telephone charges, making it a low-cost communication tool.
9. Disadvantages of Email:
   - Overload: Many people receive large volumes of email daily, leading to email overload, where important messages can be overlooked.
   - Miscommunication: The lack of non-verbal cues in email communication (such as tone of voice or body language) can sometimes lead to misunderstandings.

10. Security Risks: As email is vulnerable to hacking, phishing, and malware, users must be cautious when opening attachments or clicking on links
11. Future of Email:
    - As new communication tools and platforms emerge, email continues to evolve. Innovations include better spam filtering, enhanced security features, and improved integration with other digital tools, such as project management systems, CRMs (Customer Relationship Management), and chatbots.

Email remains a cornerstone of digital communication, with ongoing developments in features and security to adapt to the needs of users in both personal and professional contexts.

Summary

The internet is a global network that connects millions of private, public, academic, business, and government networks. It allows for the exchange of information, communication, and access to various services such as websites, social media, online shopping, and more. The internet is built on protocols like the Transmission Control Protocol/Internet Protocol (TCP/IP) that facilitate data transfer and communication between devices.

Email (electronic mail) is a method of sending and receiving messages over the internet. It allows users to send text messages, attachments (such as documents or images), and multimedia between devices. The email system relies on mail servers, which use protocols like Simple Mail Transfer Protocol (SMTP) for sending emails and Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) for receiving them.

Check your Understanding:

Question: 1

What does "HTTP" stand for?

a) Hyper Transfer Text Protocol
b) Hypertext Transfer Protocol
c) High Transfer Text Protocol
d) Hightext Transmission Protocol

Question: 2

Which of the following is the main protocol used for sending emails?

a) FTP
b) SMTP
c) IMAP
d) POP3

Question: 3

Which of the following is the correct definition of the term "URL"?

a) Universal Resource Locator
b) Uniform Resource Locator
c) Unique Resource Locator
d) Universal Retrieval Link

Question: 4

What is the full form of "IP" in IP Address?

a) Internet Protocol
b) Internal Process
c) Input Process
d) Interactive Protocol

Question: 5

What does the acronym "Wi-Fi" stand for?

a) Wireless Fidelity
b) Wireless Frequency
c) Wide Fidelity
d) Wide Frequency

Question: 6

Which of the following is the correct format for a valid email address?

a) john.doe[at]example.com
b) john.doe@examplecom
c) john.doe@example.com
d) john.doe@.com

Question: 7

What does "CC" stand for in email communication?

a) Closed Copy
b) Carbon Copy
c) Credit Copy
d) Central Copy

Question: 8

Which of the following is the purpose of the "BCC" field in an email?

a) To send a copy to the sender
b) To send a copy to all recipients openly
c) To send a copy to recipients without others knowing
d) To send a copy to the recipients' managers

Question: 9

What does the acronym "SMTP" stand for in email technology?

a) Simple Mail Transfer Protocol
b) Secure Mail Transport Protocol
c) Simple Messaging Transfer Protocol
d) Secure Mail Transfer Protocol

Question: 10

Which of the following is the primary purpose of the "Subject" field in an email?

a) To provide a summary of the email's content
b) To write the body of the email
c) To list the recipients
d) To attach files to the email

THE END

Great! Practice Regularly.