

Access 2010 VBA Macro ProgrammingVBA

Code Download

Chapter 1

```
Private Sub MyCode()  
    MsgBox "Hello World"  
End Sub
```

Chapter 2

```
Sub TestVariables()  
    stemp = "richard"  
    MsgBox VarType(stemp)  
    stemp = 4  
    MsgBox VarType(stemp)  
End Sub
```

```
Sub TestNumeric()  
    temp="richard"  
    MsgBox IsNumeric(temp)  
End Sub
```

```
Sub TestNull()  
    temp=Null  
    MsgBox IsNull(temp)  
End Sub
```

```
Dim employee(25) as String  
For n = 0 To 25  
    employee(n) = Chr(n+65)  
Next n  
For n = 0 To 25  
    MsgBox employee(n)  
Next n
```

```
Type Employee  
    Name as String  
    Salary as Currency  
    Years as Integer  
End Type
```

```
Dim temp As employee  
temp.Name = "Richard Shepherd"  
temp.Salary = 10000  
temp.Years = 5  
MsgBox temp.Name  
MsgBox temp.Salary  
MsgBox temp.Years
```

Chapter 3

```
Sub MyCode()  
    'MsgBox "Hello World"  
    x = 3 * 5  
    MsgBox x
```

```

        Call Display("my subroutine")
End Sub

Function Multiply(a, b)
    Multiply = a * b
End Function

Sub MyCode()
    MsgBox "Hello World"
    x = Multiply(3, 5)
    MsgBox x
End Sub

```

Chapter 4

```

Sub Test_If()
    If CurrentDb.Name= "C:\Temp\MyDB.accdb" Then
        MsgBox "Database path and name are correct"
    Else
        MsgBox "Database has been re-named/moved"
    End If
End Sub

x = 23
Select Case (x)
    Case (1)
        MsgBox "x=1"
    Case (23)
        MsgBox "x=23"
End Select

Sub Test_Case (Grade)
Select Case Grade
    Case 1
        MsgBox "Grade 1"
    Case 2, 3
        MsgBox "Grade 2 or 3"
    Case 4 To 6
        MsgBox "Grade 4, 5 or 6"
    Case Is > 8
        MsgBox "Grade is above 8"
    Case Else
        MsgBox "Grade not in conditional statements"
    End Select
End Sub

For n = 1 to 5
    MsgBox n
Next n

For n = 3 to 12 Step 3
    MsgBox n
Next n

For n= 10 to 1 Step -1
    MsgBox n
Next n

Sub test_loop()
    For n = 1 To 4
        For m = 1 To 5
            MsgBox "n= " & n
            MsgBox "m= " & m
        Next m
    Next n

```

```

End Sub

Sub ShowName()
    Dim oQry As QueryDef
    For Each oQry In CurrentDb.QueryDefs
        MsgBox oQry.Name
    Next oQry
End Sub

Sub test_do()
    x = 0
    Do Until x = 100
        x = x + 1
    Loop
    MsgBox x
End Sub

Sub test_do()
    x = 0
    While x < 50
        x = x + 1
    Wend
    MsgBox x
End Sub

Sub test_exit()
    For x = 1 To 100
        If x = 50 Then
            Exit For
        End If
    Next x
    MsgBox x
End Sub

```

Chapter 5

```

Sub test_sendkeys()
    x = Shell("calc.exe")
    For n = 1 To 10
        SendKeys n & "{+}", True
    Next n
    MsgBox "Press OK to close calculator"
    AppActivate "calculator"
    SendKeys "%{F4}", True
End Sub

```

Chapter 6

```

Option Compare Text
Sub test()
    MsgBox "RICHARD" Like "richard"
End Sub

```

Chapter 7

```

Sub TestDebug()
    x = 2
    Do Until x = 1
        x = x + 1
    Loop
End Sub

```

```

Sub test_for()
For n = 1 To 4
    For m = 2 To 8
        MsgBox m & " xxxxx " & n
    Next m
Next n
End Sub

```

```

Sub test_for()
For n = 1 To 4
    For m = 2 To 8
        Forms("UserForm1").Caption = n & " xxxx " & m
    Next m
Next n
End sub

```

Chapter 8

```

Sub Test_Error()
    temp = Dir("d:\*.*.")
End Sub

```

```

Sub Test_Error()
    On Error GoTo err_handler
        temp = Dir("d:\*.*.")
    Exit Sub
err_handler:
    MsgBox "The D drive is not ready" & " " & Err.Description
End Sub

```

```

Sub Test_Error()
On Error GoTo err_handler
    temp = Dir("d:\*.*.")
Exit Sub
err_handler:
If Err.Number = 71 Then
    "The D drive is not ready"
Else
    MsgBox "An error occurred"
End If
End Sub

```

```

Sub Test_Error()
On Error GoTo err_handler
    temp = Dir("d:\*.*.")
Exit Sub
err_handler:
If Err.Number = 71 Then
    MsgBox "The D drive is not ready"
Else
    MsgBox "An error occurred"
End If
Resume
End Sub

```

```

Sub Test_Error()
On Error Resume Next
    temp = Dir("d:\*.*.")
End Sub

```

```

Sub Test_Error()
    Error 71
End Sub

```

Chapter 9

```
Private Sub Mybutton_Click()  
    DoCmd.Close  
End Sub  
  
Private Sub Detail_Click()  
If Me.MyCheckBox.Value Then  
    MsgBox "Box is checked"  
Else  
    MsgBox "Box is unchecked"  
End If  
End Sub  
  
Private Sub Form_BeforeDelConfirm(Cancel As Integer, Response As Integer)  
    x = MsgBox("Are you sure that you wish to delete this record?")  
    If x = vbNo Then Cancel = True  
End Sub  
  
Private Sub Form_BeforeInsert(Cancel As Integer)  
    x = MsgBox("Are you sure that you wish to create a new record?")  
    If x = vbNo Then Cancel = True  
End Sub  
  
Private Sub Form_BeforeUpdate(Cancel As Integer)  
    x = MsgBox("Are you sure that you wish to amend this record?")  
    If x = vbNo Then Cancel = True  
End Sub  
  
Private Sub Form_Close()  
    DoCmd.OpenForm ("MyForm")  
End Sub  
  
Private Sub Form_Timer()  
    Me.Requery  
End Sub  
  
Private Sub Report_Close()  
    DoCmd.OpenReport ("MyReport")  
End Sub  
  
Private Sub Report_Timer()  
    Me.Requery  
End Sub
```

Chapter 10

```
Declare Function GetOpenFileName Lib "comdlg32.dll" Alias _  
"GetOpenFileNameA" (pOpenfilename As OPENFILENAME) As Long  
  
Private Type OPENFILENAME  
    lStructSize As Long  
    hwndOwner As Long  
    hInstance As Long  
    lpstrFilter As String  
    lpstrCustomFilter As String  
    nMaxCustFilter As Long  
    nFilterIndex As Long  
    lpstrFile As String  
    nMaxFile As Long  
    lpstrFileName As String  
    nMaxFileName As Long
```

```

lpstrInitialDir As String
lpstrTitle As String
flags As Long
nFileOffset As Integer
nFileExtension As Integer
lpstrDefExt As String
lCustData As Long
lpfnHook As Long
lpTemplateName As String
End Type

```

```

Private Sub ShowFileDialog()
Dim MyFile As OPENFILENAME
Dim ReturnValue As Long
Dim strFilter As String
MyFile.lStructSize = Len(MyFile)
MyFile.hwndOwner = Me.hwnd
strFilter = "Text Files (*.txt)" & Chr(0) & "*.TXT" & Chr(0)
MyFile.lpstrFilter = strFilter
MyFile.nFilterIndex = 1
MyFile.lpstrFile = String(257, 0)
MyFile.nMaxFile = Len(MyFile.lpstrFile) - 1
MyFile.lpstrFileName = MyFile.lpstrFile
MyFile.nMaxFileName = MyFile.nMaxFile
MyFile.lpstrInitialDir = "C:\\"
MyFile.lpstrTitle = "Select a File"
MyFile.flags = 0
ReturnValue = GetOpenFileName(MyFile)
If ReturnValue = 0 Then
    MsgBox "Cancel Button was pressed"
Else
    MsgBox MyFile.lpstrFile
End If
End Sub

```

```

Private Sub Command0_Click()
ShowFileDialog
End Sub

```

```

Private Declare Function ChooseColor _
Lib "comdlg32.dll" Alias "ChooseColorA" _
(pChoosecolor As COLORSTRUC) As Long
Private Type COLORSTRUC
lStructSize As Long
hwnd As Long
hInstance As Long
rgbResult As Long
lpCustColors As String
Flags As Long
lCustData As Long
lpfnHook As Long
lpTemplateName As String
End Type

```

```

Private Sub ShowColorDialog()
Dim x As Long, CSel As COLORSTRUC, CustColor(16) As Long
CSel.lStructSize = Len(CSel)
CSel.hwnd = Me.hwnd
CSel.Flags = &H80
CSel.lpCustColors = String$(16 * 4, 0)
x = ChooseColor(CSel)
If x = 0 Then
    MsgBox "The user pressed Cancel"
Else
    MsgBox CSel.rgbResult
End If
End Sub

```

Chapter 11

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
<ribbon startFromScratch="true">
</ribbon>
</customUI>

<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
    <ribbon>
        <tabs>
            <tab id="customTab" label="Contoso" insertAfterMso="TabHomeAccess">
                <group idMso="GroupClipboard" />
                <group idMso="GroupFontAccess" />
                <group id="customGroup" label="Contoso Tools">
                    <button id="customButton1" label="ConBold"
size="large" onAction="conBoldSub" imageMso="Bold" />
                    <button id="customButton2" label="ConItalic"
size="large" onAction="conItalicSub" imageMso="Italic" />
                    <button id="customButton3" label="ConUnderline"
size="large" onAction="conUnderlineSub" imageMso="Underline" />
                </group>
                <group idMso="GroupEnterDataAlignment" />
                <group idMso="GroupEnterDataNumber" />
                <group idMso="GroupQuickFormatting" />
            </tab>
        </tabs>
    </ribbon>
</customUI>

Sub conBoldSub(control As IRibbonControl)
    Screen.ActiveControl.FontBold = True
End Sub

Sub MyLabel(control As IRibbonControl, ReturnValue As Variant)
    ReturnValue = "My new title"
End Sub

<group id="Group1" label="My Control">
<checkBox id="Checkbox1"
label="Checkbox sample"
onAction="MyCheckbox"
/>

<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
    <ribbon>
        <tabs>
            <tab id="customTab" label="Contoso"
insertAfterMso="TabHomeAccess">
                <group idMso="GroupClipboard" />
                <group idMso="GroupFontAccess" />
                <group id="customGroup" getLabel="MyLabel">
                    <button id="customButton1" label="ConBold"
size="large" onAction="conBoldSub" imageMso="HappyFace" />
                    <button id="customButton2" label="ConItalic"
size="large" onAction="conItalicSub" imageMso="Italic" />
                    <button id="customButton3" label="ConUnderline"
size="large" onAction="conUnderlineSub" imageMso="Underline" />
                </group>
```

```

        <group idMso="GroupEnterDataAlignment" />
        <group idMso="GroupEnterDataNumber" />
        <group idMso="GroupQuickFormatting" />
        <group id="Group1" label="My Control">
        <checkBox id="Checkbox1"
        label="Checkbox sample"
        onAction="MyCheckbox"
        />
        </group>
    </tab>
</tabs>
</ribbon>
</customUI>

Sub MyCheckbox(control As IRibbonControl, pressed As Boolean)
    MsgBox pressed
End Sub

<editBox id="Editbox1"
label="Editbox sample"
onChange="MyChange"/>

<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
<ribbon>
    <tabs>
        <tab id="customTab" label="Contoso"
            insertAfterMso="TabHomeAccess">
            <group idMso="GroupClipboard" />
            <group idMso="GroupFontAccess" />
            <group id="customGroup" getLabel="MyLabel">
                <button id="customButton1" label="ConBold"
                size="large" onAction="conBoldSub" imageMso="HappyFace" />
                <button id="customButton2" label="ConItalic"
                size="large" onAction="conItalicSub" imageMso="Italic" />
                <button id="customButton3" label="ConUnderline"
                size="large" onAction="conUnderlineSub" imageMso="Underline" />
            </group>
            <group idMso="GroupEnterDataAlignment" />
            <group idMso="GroupEnterDataNumber" />
            <group idMso="GroupQuickFormatting" />
            <group id="customGroup1" label="My Control">
            <checkBox id="Checkbox1"
            label="Checkbox sample"
            onAction="MyCheckbox"
            />
            <editBox id="Editbox1"
            label="Editbox sample"
            onChange="MyChange"
            />
            </group>
        </tab>
    </tabs>
</ribbon>
</customUI>

Sub MyChange(control As IRibbonControl, text As String)
    MsgBox text
End Sub

<comboBox

```

```

id="MyCombo" label="My Combo"
onChange ="OnChange"
getItemCount ="GetItemCount"
getItemLabel ="GetItemLabel"
/>

```

```

<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
    <ribbon>
        <tabs>
            <tab id="customTab" label="Contoso"
insertAfterMso="TabHomeAccess">
                <group idMso="GroupClipboard" />
                <group idMso="GroupFontAccess" />
                <group id="customGroup" getLabel="MyLabel">
                    <button id="customButton1" label="ConBold"
size="large" onAction="conBoldSub" imageMso="HappyFace" />
                    <button id="customButton2" label="ConItalic"
size="large" onAction="conItalicSub" imageMso="Italic" />
                    <button id="customButton3" label="ConUnderline"
size="large" onAction="conUnderlineSub" imageMso="Underline" />
                </group>
                <group idMso="GroupEnterDataAlignment" />
                <group idMso="GroupEnterDataNumber" />
                <group idMso="GroupQuickFormatting" />
                <group id="customGroup1" label="My Control">
                    <checkBox id="Checkbox1"
label="Checkbox sample"
onAction="MyCheckbox"
/>
                    <editBox id="Editbox1"
label="Editbox sample"
onChange="MyChange"
/>
                    <comboBox
id="MyCombo"
label="My Combo"
onChange ="OnChange"
getItemCount ="GetItemCount"
getItemLabel ="GetItemLabel"
/>
                </group>
            </tab>
        </tabs>
    </ribbon>
</customUI>

```

```

Sub GetItemCount(control As IRibbonControl, ByRef returnedVal)
returnedVal = 4
End Sub
Sub GetItemLabel(control As IRibbonControl, index As Integer, ByRef returnedVal)
ComboArray(0) = "Option1"
ComboArray(1) = "Option2"
ComboArray(2) = "Option3"
ComboArray(3) = "Option4"
returnedVal = ComboArray(index)
End Sub

Sub OnChange(control As IRibbonControl, text As String)
MsgBox text
End Sub

```

Chapter 12

```
SELECT Customers.Company, Orders.[Order Date], Orders.[Ship Name]
FROM Customers RIGHT JOIN Orders ON Customers.ID = Orders.[Customer ID]
WHERE (((Orders.[Order Date])>#1/1/2006#));
```

```
Sub TestQuery()
Dim ReSet As Recordset
Set ReSet = CurrentDb.OpenRecordset("MyQuery")
Do Until ReSet.EOF
    MsgBox ReSet!Company
    ReSet.MoveNext
Loop
End Sub
```

```
Sub TestQuery()
Dim ReSet As Recordset
Set ReSet = CurrentDb.OpenRecordset("MyQuery")
Do Until ReSet.EOF
    If ReSet![Ship Name] = "Karen Toh" Then
        ReSet.Edit
        ReSet![Ship Name] = "Unknown"
        ReSet.Update
    End If
    ReSet.MoveNext
Loop
End Sub
```

```
Sub TestQuery()
Dim ReSet As Recordset
Set ReSet = CurrentDb.OpenRecordset("MyQuery")
ReSet.AddNew
ReSet!Company = "Richard"
ReSet![Order Date] = "01-Aug-2006"
ReSet![Ship Name] = "Richard Shepherd"
ReSet.Update
End Sub
```

```
Select MyRef,Monday,Tuesday,0 as Wednesday,0 as Thursday from MyTable1
Union
Select MyRef,0 as Monday,0 as Tuesday, Wednesday, Thursday from MyTable2
```

```
Select MyRef,sum(Monday),sum(Tuesday),sum(Wednesday),sum(Thursday) from (
Select MyRef,Monday,Tuesday,0 as Wednesday,0 as Thursday from MyTable1
Union
Select MyRef,0 as Monday,0 as Tuesday, Wednesday, Thursday from MyTable2
) group by MyRef
```

```
Select MyRef,sum(Monday),sum(Tuesday),sum(Wednesday),sum(Thursday) from
MyUnionQuery group by MyRef
```

```
Sub DeleteQuery()
DoCmd.SetWarnings False
DoCmd.OpenQuery "MyDeleteQuery"
DoCmd.SetWarnings True
End Sub
```

```
Sub MakeTableQuery()
DoCmd.SetWarnings False
DoCmd.OpenQuery "MyMakeTableQuery"
DoCmd.SetWarnings True
End Sub
```

```
INSERT INTO NewCustomers ( Company, CustomerName )
```

```

SELECT Company, CustomerName
FROM MyTable

Sub MakeTableQuery()
DoCmd.SetWarnings False
DoCmd.OpenQuery "MyAppendQuery"
DoCmd.SetWarnings True
End Sub

Src = " INSERT INTO NewCustomers "
Src = Src & "SELECT * FROM MyTable"
CurrentDb.Execute Src

UPDATE MyTable SET CustomerName = "Richard"
WHERE (((CustomerName)="Shepherd"))

UPDATE MyTable SET CustomerName = "Richard", Company = "MGH"
WHERE (((CustomerName)="Shepherd") AND ((Company)="RBS"))

Sub MakeTableQuery()
DoCmd.SetWarnings False
DoCmd.OpenQuery "MyUpdateQuery"
DoCmd.SetWarnings True
End Sub

Src = " UPDATE MyTable SET CustomerName = 'Richard' "
Src = Src & "WHERE (((CustomerName)='Shepherd'))"

Function CheckUpperCase(Target As String) As Boolean
Dim Flag As Integer
For n = 1 To Len(Target)
    If Asc(Mid(Target, n, 1)) >= 97 Then
        CheckUpperCase = False
        Exit Function
    End If
Next n
CheckUpperCase = True
End Function

```

Chapter 13

No code

Chapter 14

```

DoCmd.TransferSpreadsheet TransferType:=acImport,TableName:="MyTable", _
FileName:="C:\MyTable",Range:="A1:G10"

Sub ShowName()
Dim w As TableDef
For Each w In CurrentDb.TableDefs
    MsgBox w.Name
Next w
End Sub

Sub ViewData()
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("Employees")
Do Until RecSet.EOF
    MsgBox RecSet![Last Name]
    MsgBox RecSet![First Name]
    RecSet.MoveNext
Loop
RecSet.Close
Set RecSet = Nothing

```

```

End Sub

Sub EditData()
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("Employees")
Do Until RecSet.EOF
    If RecSet![First Name] = "Nancy" Then
        RecSet.Edit
        RecSet![Company] = "Northwind"
        RecSet.Update
    End If
    RecSet.MoveNext
Loop
RecSet.Close
Set RecSet = Nothing
End Sub

Sub AddData()
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("Employees")
RecSet.AddNew
RecSet!Company = "MyCompany"
RecSet![Last Name] = "Shepherd"
RecSet![First Name] = "Richard"
RecSet.Update
RecSet.Close
Set RecSet = Nothing
End Sub

Sub EditData()
Dim RecSet As Recordset, WrkSp as Workspace
Set RecSet = CurrentDb.OpenRecordset("Employees")
Set WrkSp = DBEngine.Workspaces(0)
WrkSp.BeginTrans
Do Until RecSet.EOF
    If RecSet![First Name] = "Nancy" Then
        RecSet.Edit
        RecSet![Company] = "Northwind"
        RecSet.Update
    End If
    RecSet.MoveNext
Loop
If MsgBox("Save all changes?", vbQuestion + vbYesNo) = vbYes Then
    WrkSp.CommitTrans
Else
    WrkSp.Rollback
End If
RecSet.Close
WrkSp.close
Set RecSet = Nothing
Set WrkSp = Nothing
End Sub

Dim Db As Database
Dim MyTable As TableDef, MyField As Field
Set Db = CurrentDb
Set MyTable = Db.TableDefs("Employees")
Set MyField = MyTable.Fields("Company")
MsgBox MyField.Size

```

Chapter 15

```

Sub EditData()
Dim RecSet As Recordset, WrkSp as Workspace
Set RecSet = CurrentDb.OpenRecordset("Employees")

```

```

Set WrkSp = DBEngine.Workspaces(0)
WrkSp.BeginTrans
Do Until RecSet.EOF
    If RecSet![First Name] = "Nancy" Then
        RecSet.Edit
        RecSet![Company] = "Northwind"
        RecSet.Update
    End If
    RecSet.MoveNext
Loop
If MsgBox("Save all changes?", vbQuestion + vbYesNo) = vbYes Then
    WrkSp.CommitTrans
Else
    WrkSp.Rollback
End If
RecSet.Close
WrkSp.close
Set RecSet = Nothing
Set WrkSp = Nothing
End Sub

For Each frm In Forms
    MsgBox frm.Name
Next frm

Dim Prt As Printer
For Each Prt In Printers
    MsgBox Prt.DeviceName
Next Prt

Dim Reff As Reference
For Each Reff In References
    MsgBox Reff.Name
Next Reff

Dim Rep As Report
For Each Rep In Reports
    MsgBox Rep.Name
Next Rep

If Application.Version <> "12.0" Then
    MsgBox "You need a later version of Access to run this application",
vbCritical
    Quit
End If

Private Sub Detail_Click()
    MsgBox Me.ActiveControl.Name
End Sub

Sub CreateTable()
Dim MyTable As TableDef
Set MyTable = CurrentDb.CreateTableDef("NewTable")
MyTable.Fields.Append MyTable.CreateField("LastName", dbText, 100)
CurrentDb.TableDefs.Append MyTable
End Sub

Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("MyTable")

Dim Qd As QueryDef

```

```
For Each Qd In CurrentDb.QueryDefs
    MsgBox Qd.Name & " " & Qd.SQL
Next Qd
```

```
Dim Td As TableDef
For Each Td In CurrentDb.TableDefs
    MsgBox Td.Name & " " & Td.Fields.Count
Next Td
```

```
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("MyTable")
Do Until RecSet.EOF
    MsgBox RecSet![MyField]
    RecSet.MoveNext
Loop
```

```
RecSet.AddNew
RecSet![MyField1] = "MyData1"
RecSet![MyField2] = "MyData2"
RecSet.Update
```

```
RecSet.Edit
RecSet![MyField1] = "MyNewData1"
RecSet![MyField2] = "MyNewData2"
RecSet.Update
```

```
With RecSet.Fields(0)
    MsgBox .Name
    MsgBox .Type
    MsgBox .Value
    MsgBox .Size
End With
```

```
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("MyTable", dbOpenDynaset)
RecSet.FindFirst ("MyField='Data'")
MsgBox RecSet!MyField
RecSet.close
Set RecSet=Nothing
```

```
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("MyTable", dbOpenDynaset)
RecSet.FindLast ("MyField='Data'")
MsgBox RecSet!MyField
RecSet.close
Set RecSet=Nothing
```

```
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("MyTable", dbOpenDynaset)
RecSet.FindNext ("MyField='Data'")
MsgBox RecSet!MyField
RecSet.close
Set RecSet=Nothing
```

```
Dim RecSet As Recordset
Set RecSet = CurrentDb.OpenRecordset("MyTable", dbOpenDynaset)
RecSet.FindPrevious ("MyField='Data'")
MsgBox RecSet!MyField
RecSet.close
Set RecSet=Nothing
```

```
RecSet.FindFirst ("MyField='Data'")
MsgBox RecSet.NoMatch
```

Chapter 16

```
DoCmd.TransferSpreadsheet _
(TransferType, SpreadsheetType, TableName, FileName, HasFieldNames, Range, UseOA)
```

```
DoCmd.TransferText _
(TransferType, SpecificationName, TableName, FileName, HasFieldNames, _
HTMLTableName, CodePage)
```

Chapter 17

```
Sub Test_Word()
Dim oWd As Word.Application, oWdoc As Word.Document
Set oWd = CreateObject("Word.Application")
Set oWdoc = oWd.Documents.Add
oWdoc.Sections(1).Range.Text = "My new Word Document"
oWdoc.SaveAs ("MyTest")
oWdoc.Close
oWd.Quit
Set oWdoc = Nothing
Set oWd = Nothing
End Sub
```

```
Sub Test_Word()
Dim oWd As Word.Application, oWdoc As Word.Document
Dim orx As Word.Range, ot As Word.Table
Set oWd = CreateObject("Word.Application")
Set oWdoc = oWd.Documents.Add
Set orx = oWdoc.Range
Set ot = oWdoc.Tables.Add(orx, 4, 5)
ot.Cell(1, 1).Range.Text = "test"
oWdoc.SaveAs ("MyTest")
oWdoc.Close
oWd.Quit
Set oWdoc = Nothing
Set oWd = Nothing
End Sub
```

```
Sub Test_Outlook()
Dim oFolder As Outlook.MAPIFolder
Dim oItem As Outlook.MailItem
Dim oOutlook As New Outlook.Application
Dim MoOutlook As Outlook.Namespace
Set MoOutlook = oOutlook.GetNamespace("MAPI")
Set oFolder = MoOutlook.GetDefaultFolder(olFolderOutbox)
Set oItem = oFolder.Items.Add(olMailItem)
With oItem
.Reciipients.Add ("richard.shepherd@anywhere.com")
.Subject = "Test Access Email"
.Body = "This is a test of an email from Access VBA"
.Importance = olImportanceHigh
.Send
End With
Set oItem = Nothing
Set oFolder = Nothing
End Sub
```

```
Sub Test_Access()
Dim oAApp As Access.Application, oRSet As DAO.Recordset
```

```

Set oApp = CreateObject("Access.Application")
oApp.OpenCurrentDatabase "C:\Temp\TestAccess.accdb"
Set oRSet = oApp.CurrentDb.OpenRecordset("MyTable")
oRSet.AddNew
oRSet!MyText = "Test String"
oRSet.Update
oRSet.Close
Set oRSet = Nothing
Set oApp = Nothing
End Sub

```

Chapter 18

```

Sub CreateGraphOnReport()
Dim rep As Report
Dim ctl As control
Dim ch As Graph.Chart
Dim RepName As String
DoCmd.OpenReport "rptTemplate", acDesign, , , , acHidden
Set rep = CreateReport()
Set ctl = CreateReportControl(rep.Name, acObjectFrame, acDetail)
ctl.OleData = Reports!rptTemplate!Graph0.OleData
DoCmd.Restore
DoCmd.Close acReport, "rptTemplate"
RepName = rep.Name
rep.OLEUnbound0.RowSource = "tblChart"
rep.OLEUnbound0.RowSourceType = "Table/Query"
rep.OLEUnbound0.ColumnHeads = True
rep.OLEUnbound0.Top = 100
rep.OLEUnbound0.Left = 100
DoCmd.Restore
DoCmd.Save acDefault
DoCmd.Close acDefault
DoCmd.Rename "rptMyGraph", acReport, RepName
DoCmd.OpenReport "rptMyGraph", acViewDesign
DoCmd.Restore
Set ch = Reports("rptMyGraph").OLEUnbound0.Object
ch.HasTitle = True
ch.ChartTitle.Text = "My Chart"
ch.ChartType = xl3DPie
DoCmd.Save acDefault
DoCmd.Close acDefault
DoCmd.OpenReport "rptMyGraph", acViewReport
End Sub

```

```

Dim ch As Graph.Chart
Set ch = Me.Graph0.Object

ch.SeriesCollection(1).HasDataLabels = True
ch.SeriesCollection(1).Explosion = 8
ch.SeriesCollection(1).DataLabels.ShowValue = True
ch.SeriesCollection(1).DataLabels.ShowCategoryName = True
ch.SeriesCollection(1).Points(1).Interior.Color = QBColor(12)

```

```

Sub OutputChart()
Dim ch As Graph.Chart
Set ch = Me.Graph0.Object
ExportFile = Access.CodeProject.Path & "\" MyChart.gif"
ch.Export Filename:=ExportFile, FilterName:="GIF"
End Sub

```

Chapter 19

```

DBEngine.RegisterDatabase "MyDSN", "SQL Server Native Client 10.0", True, _
"MyServer" _
& vbCr & "Database=MyDatabase"

Sub ADOExtract()
Dim RsADO As ADODB.Recordset, RsAccess As Recordset
Dim Cnct As String, Cnct1 as String, Cnct2 as String
Set Connection = New ADODB.Connection
Cnct1 = "Provider=SQLOLEDB;Driver={SQL Server Native Client 10.0};"
Cnct2="Server=MyServer;Database=MyDatabase;Uid=MyId;Pwd=MyPassword;"
Cnct=Cnct1 & Cnct2
Connection.CommandTimeout = 60
Connection.Open ConnectionString:=Cnct
Set RsAccess = CurrentDb.OpenRecordset("MyDestinationTable")
Set RsADO = New ADODB.Recordset
With RsADO
.Open Source="select * from MySourceTable", _
ActiveConnection:=Connection, CursorType:=adOpenStatic
Do Until RsADO.EOF
RsAccess.AddNew
RsAccess!Field1 = RsADO!Field1
RsAccess!Field2 = RsADO!Field2
RsAccess.Update
RsADO.MoveNext
Loop
End With
Set RsADO = Nothing
Set RsAccess = Nothing
End Sub

OLEDB;DSN=MyDSN;Uid=MyId;Pwd=MyPassword;

Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\myFolder\myAccess2007file.accdb;

Sub FastTransferofData()
Dim Cnct As String, Ccnt1 as String, Ccnt2 as String
Cnct1 = "ODBC;Provider=MDASQL;Driver={SQL Server};"
Ccnt2 = "Server=MyServer;Database=MyDatabase;Uid=MyId;Pwd=MyPassword;"
Ccnt = Ccnt1 & Ccnt2
CurrentDb.Execute "insert into DestTble select * from [" & Cnct & "].SrceTble"
End Sub

```

Chapter 20

```

Private Declare Function GetDiskFreeSpaceEx Lib "kernel32" _
Alias "GetDiskFreeSpaceExA" (ByVal lpDirectoryName As _
String, lpFreeBytesAvailableToCaller As Currency, _
lpTotalNumberOfBytes As Currency, lpTotalNumberOfFreeBytes _
As Currency) As Long

Sub Test_Api()
Dim x as Double
Dim FreeBytesAvailableToCaller As Currency, TotalNumberOfBytes As _
Currency, TotalNumberOfFreeBytes As Currency
x = GetDiskFreeSpaceEx("c:\", FreeBytesAvailableToCaller, _
TotalNumberOfBytes, TotalNumberOfFreeBytes)
MsgBox "Total Space " & Format(TotalNumberOfBytes * 10000, "#,##0")
MsgBox "Free Space " & Format(TotalNumberOfFreeBytes * 10000, "#,##0")
End Sub

Private Declare Function GetPrivateProfileString Lib "kernel32" Alias _
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal _
lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As _
String, ByVal nSize As Long, ByVal lpFileName As String) As Long
Private Declare Function WritePrivateProfileString Lib "kernel32" Alias _

```

```

"WritePrivateProfileString" (ByVal lpApplicationName As String, ByVal _
lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As _
Long

Sub Test_INI()
x = WritePrivateProfileString("Parameters", "Path", "C:\temp\", "myini.ini")
s$ = Space$(256)
x = GetPrivateProfileString("Parameters", "Path", "", s$, 256, "myini.ini")
MsgBox s$
MsgBox x
End Sub

Private Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) _
As Integer

Sub Test_Key()
x = 0
Do Until x = 1
If GetKeyState(&H9) < 0 Then x = 1
DoEvents
Loop
MsgBox "You pressed the TAB key"
End Sub

Public Declare Function PlaySound Lib "winmm.dll" Alias "PlaySoundA" _
(ByVal lpszName As String, ByVal hModule As Long, ByVal dwFlags As _
Long) As Long

Sub test_sound()
x = PlaySound("c:\windows\media\windows exclamation.wav", 0, 0)
x = PlaySound("c:\windows\media\tada.wav", 0, 0)
End Sub

```

Chapter 21

```

Private mName As String

Public Property Let PName(vdata As String)
    mName = vdata
End Property

Public Property Get PName() As String
    PName = mName
End Property

Private mPNames As Collection

Private Sub Class_Initialize()
Dim objPName As New MyPName
Dim MyConn As New ADODB.Connection, ReSet As New ADODB.Recordset
Dim ConStr As String
ConStr = "Driver={Microsoft Access Driver (*.mdb, *.accdb)};" & _
"Dbq=MyTable.accdb;" & _
"DefaultDir=C:\MyPath\" & _
"Uid=Admin;Pwd=;"
MyConn.ConnectionString = ConStr
MyConn.Open
Set ReSet = MyConn.Execute("select * from MyTable")
Set mPNames = New Collection
Do Until ReSet.EOF
    Set objPName = New MyPName
    objPName.PName = ReSet!MyName
    ReSet.MoveNext
    mPNames.Add objPName

```

```

Loop
End Sub

```

```

Public Function Item(Index As Integer) As Pname
    Set Item = mPnames.Item(Index)
End Function

```

```

Public Property Get Count() As Long
    Count = mPnames.Count
End Property

```

```

Sub test_class()
Dim pn As New MyPNames, n As Integer
MsgBox pn.Count
For n = 1 To pn.Count
    pn.Item(n).Pname
Next n
End Sub

```

```

Sub test_class()
Static pn As New MyPNames, n As Integer
MsgBox pn.Count
For n = 1 To pn.Count
    MsgBox pn.Item(n).Pname
Next n
End Sub

```

```

Sub test_class()
Dim pn As MyPNames, n As Integer
Set pn = New MyPNames
MsgBox pn.Count
For n = 1 To pn.Count
    MsgBox pn.Item(n).Pname
Next n
End Sub

```

Chapter 22

```

Private Sub Form_Open(Cancel As Integer)
Dim VIncr As Long, HIncr As Long, Spd As Integer
Spd = 15
VIncr = Spd
HIncr = Spd
Do Until 1 = 2
    If Me.MyLabel.Left + HIncr < 0 Then HIncr = Spd
    If Me.MyLabel.Left > Me.InsideWidth - Me.MyLabel.Width Then HIncr = -Spd
    If Me.MyLabel.Top + VIncr < 0 Then VIncr = Spd
    If Me.MyLabel.Top > Me.InsideHeight - Me.MyLabel.Height Then VIncr = -Spd
    Me.MyLabel.Left = Me.MyLabel.Left + HIncr
    Me.MyLabel.Top = Me.MyLabel.Top + VIncr
    Me.Repaint
Loop
End Sub

```

Chapter 23

```

Public Declare Function GetUserName Lib "advapi32.dll" Alias _
    "GetUserNameA" (ByVal lpBuffer As String, nSize As Long) As Long

Public Function ReturnUserName()
Dim strUser As String, X As Integer
strUser = Space$(256)
X = GetUserName(strUser, 256)
strUser = RTrim(strUser)

```

```
ReturnUserName = Left(strUser, Len(strUser) - 1)
End Function
```

```
Sub CheckUser()
Dim RcSet As Recordset
Set RcSet = CurrentDb.OpenRecordset _
("select * from AuthorizedUsers where UserName='" & ReturnUserName & "'")
If RcSet.RecordCount = 0 Then
    MsgBox "You are not authorized to use this application", vbInformation
    DoCmd.Quit
Else
    MsgBox "Welcome to " & ReturnUserName & ",vbInformation"
End If
End Sub
```

Chapter 24

```
Sub DisableStartupProperties()
    ChangeProperty "StartupShowDBWindow", dbBoolean, False
    ChangeProperty "StartupShowStatusBar", dbBoolean, False
    ChangeProperty "AllowBuiltinToolbars", dbBoolean, False
    ChangeProperty "AllowFullMenus", dbBoolean, False
    ChangeProperty "AllowBreakIntoCode", dbBoolean, False
    ChangeProperty "AllowSpecialKeys", dbBoolean, False
    ChangeProperty "AllowBypassKey", dbBoolean, False
    ChangeProperty "AllowShortcutMenus", dbBoolean, False
End Sub
Function ChangeProperty(strPropName As String, _
varPropType As Variant, varPropValue As Variant) As Integer
    Dim dbs As Database, prp As Property
    Const conPropNotFoundError = 3270
    Set dbs = CurrentDb
    On Error GoTo Change_Err
    dbs.Properties(strPropName) = varPropValue
    ChangeProperty = True
Change_Bye:
    Exit Function
Change_Err:
    If Err = conPropNotFoundError Then
        Set prp = dbs.CreateProperty(strPropName, varPropType, varPropValue)
        dbs.Properties.Append prp
        Resume Next
    Else
        ChangeProperty = False
        Resume Change_Bye
    End If
End Function
```

```
<customUI
xmlns="http://schemas.microsoft.com/office/2006/01/customui">
<ribbon startFromScratch="true">
</ribbon>
</customUI>
```

```
Sub EnableStartupProperties()
    ChangeProperty "StartupShowDBWindow", dbBoolean, True
    ChangeProperty "StartupShowStatusBar", dbBoolean, True
    ChangeProperty "AllowBuiltinToolbars", dbBoolean, True
    ChangeProperty "AllowFullMenus", dbBoolean, True
    ChangeProperty "AllowBreakIntoCode", dbBoolean, True
    ChangeProperty "AllowSpecialKeys", dbBoolean, True
    ChangeProperty "AllowBypassKey", dbBoolean, True
    ChangeProperty "AllowShortcutMenus", dbBoolean, True
End Sub
```

```

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = 191 And Shift = 1 Then
    EnableStartupProperties
    MsgBox "Security Disabled", vbInformation
End If
End Sub

```

Chapter 25

```

Public Declare Function GetUserName Lib "advapi32.dll" Alias _
"GetUserNameA" (ByVal lpBuffer As String, nSize As Long) As Long

Public Function ReturnUserName()
Dim strUser As String, X As Integer
strUser = Space$(256)
X = GetUserName(strUser, 256)
strUser = RTrim(strUser)
ReturnUserName = Left(strUser, Len(strUser) - 1)
End Function

Dim RecSet As Recordset, Temp As Variant
Set RecSet = CurrentDb.OpenRecordset("select * from Customers where ID=" & Me.ID)
Temp = RecSet!AuditTrail
Temp = Temp & "Edit" & "|" & ReturnUserName & "|" & Now() & "|"
CurrentDb.Execute ("update Customers set AuditTrail='" & Temp & "' where ID=" & Me.ID)

Dim Temp As Variant
Temp = Temp & "Edit" & "|" & ReturnUserName & "|" & Now() & "|"
CurrentDb.Execute ("update Customers set AuditTrail='" & Temp & "' where ID=" & Me.ID)

Private Sub Form_AfterUpdate()
Dim RecSet As Recordset, Temp As Variant
Set RecSet = CurrentDb.OpenRecordset("select * from Customers where ID=" & Me.ID)
Temp = RecSet!AuditTrail
Temp = Temp & "Edit" & "|" & ReturnUserName & "|" & Now() & "|" & Me.First_Name & "|" & Me.Last_Name & "|" & Me.E_mail_Address
CurrentDb.Execute ("update Customers set AuditTrail='" & Temp & "' where ID=" & Me.ID)
End Sub

```

Chapter 26

```

Sub CreateQuery()
Dim Qd As New QueryDef, Qds As QueryDefs
Set Qds = CurrentDb.QueryDefs
Qd.Name = "MyNewQuery"
Qd.SQL = "Select * from orders"
Qds.Append Qd
Application.RefreshDatabaseWindow
Set Qd = Nothing
Set Qds = Nothing
End Sub

Sub RemoveQuery()
CurrentDb.QueryDefs.Delete "MyNewQuery"
Application.RefreshDatabaseWindow
End Sub

Sub ChangeQuery()
CurrentDb.QueryDefs("MyNewQuery").SQL = _
"update orders set customer ='Company Unknown' where
customer='Company D'"

```

```

Application.RefreshDatabaseWindow
End Sub

Sub ChangeQuery()
CurrentDb.QueryDefs("MyNewQuery").SQL = _
"delete * from orders where customer='Customer D'"
Application.RefreshDatabaseWindow
End Sub

Sub SetTimeout()
CurrentDb.QueryDefs("MyNewQuery").ODBCTimeout = 120
End Sub

```

Chapter 27

```

Sub SearchQueries()
Dim RecSet As Recordset, Qdef As QueryDef, SearchStr As String
CurrentDb.Execute "delete * from tblSearchResults"
Set RecSet = CurrentDb.OpenRecordset("tblSearchResults")
SearchStr = "orders"
For Each Qdef In CurrentDb.QueryDefs
    If InStr(Qdef.SQL, SearchStr) Then
        RecSet.AddNew
        RecSet!QueryName = Qdef.Name
        RecSet.Update
    End If
Next Qdef
Set RecSet = Nothing
Set Qdef = Nothing
End Sub

Sub SearchReplaceQueries()
Dim QName As String, SearchStr As String, ReplaceStr As String
Dim Temp As String
QName = "MyNewQuery"
SearchStr = "Company D"
ReplaceStr = "Company B"
Temp = CurrentDb.QueryDefs(QName).SQL
For n = 1 To Len(Temp)
    X = InStr(n, Temp, SearchStr)
    If X Then
        Temp = Left(Temp, X - 1) & ReplaceStr & Mid(Temp, X + Len(ReplaceStr))
    End If
Next n
CurrentDb.QueryDefs(QName).SQL = Temp
End Sub

```

Chapter 28

```

Function TestDateAdd(Target, Months As Integer)
Dim Temp As String
If IsDate(Target) = False Then
    TestDateAdd = "Invalid date"
    Exit Function
End If
Temp = Right(Target, 2)
If Temp <= 29 Then
    Temp = "20" & Temp
Else
    Temp = "19" & Temp
End If
Target = Month(Target) & "/" & Temp
TestDateAdd = DateAdd("m", Months + 1, Target)
TestDateAdd = DateAdd("d", -1, TestDateAdd)

```

```
End Function
```

```
Function AccessWait(Target As Long)
TimeWait = DateAdd("s", Target, Time)
Do Until Time >= TimeWait
Loop
End Function
```

Chapter 29

```
Sub TableStatistics()
Dim RecSet As Recordset, RecSet1 As Recordset, TDef As TableDef
CurrentDb.Execute "delete * from tblTableStatistics" 'Optional
Set RecSet = CurrentDb.OpenRecordset("tblTableStatistics")
For Each TDef In CurrentDb.TableDefs
    Set RecSet1 = CurrentDb.OpenRecordset(TDef.Name)
    RecSet.AddNew
    RecSet!TableName = TDef.Name
    RecSet!TableSize = RecSet1.RecordCount
    RecSet!TimeStamp = Now()
    RecSet.Update
Next TDef
Set RecSet = Nothing
Set RecSet1 = Nothing
End Sub
```

Chapter 30

```
Sub CreateTenRows()
Dim temp As String
fnum = FreeFile
Open "C:\MyLargeFile.txt" For Input As fnum
fnum1 = FreeFile
Open "c:\FirstTenRows.txt" For Output As fnum1
For n = 1 To 10
    Line Input #fnum, temp
    Print #fnum1, temp
Next n
Close fnum
Close fnum1
End Sub
```

```
Sub GetMyBigFile()
Dim temp As String, temp1 As String
fnum = FreeFile
Open "C:\MyBigFile.txt" For Input As fnum
fnum1 = FreeFile
Open "c:\MySmallFile.txt" For Output As fnum1
Do While Not EOF(fnum)
    Line Input #fnum, temp
    temp1 = ""
    For m = 1 To Len(temp)
        If Mid(temp, m, 1) = "~" Then
            temp1 = temp1 & ","
        ElseIf Mid(temp, m, 1) = "" Then
        Else
            temp1 = temp1 & Mid(temp, m, 1)
        End If
    Next m
    Print #fnum1, temp1
Loop
Close fnum
Close fnum1
End Sub
```

```

Sub CreateSplitFiles()
Dim temp As String, temp1 As String, temp2 As String, n As Long
n = 1
fnum = FreeFile
Open "C:\
Open "c:\SmallFile1.txt" For Output As fnum1
fnum2 = FreeFile
Open "c:\SmallFile2.txt" For Output As fnum2
Do While Not EOF(fnum)
    Line Input #fnum, temp
    temp1 = ""
    For m = 1 To Len(temp)
        If Mid(temp, m, 1) = "~" Then
            temp1 = temp1 & ","
        ElseIf Mid(temp, m, 1) = "'" Then
        Else
            temp1 = temp1 & Mid(temp, m, 1)
        End If
    Next m
    If n = 1 Then
        Print #fnum2, temp1
    End If
    If n < 3500000 Then
        Print #fnum1, temp1
    Else
        Print #fnum2, temp1
    End If
    n = n + 1
Loop
Close fnum
Close fnum1
Close fnum2
End Sub

```

Chapter 31

```

Sub CreateTable()
Dim MyTable As TableDef
Set MyTable = CurrentDb.CreateTableDef("NewTable")
MyTable.Fields.Append MyTable.CreateField("TextField", dbText, 100)
MyTable.Fields.Append MyTable.CreateField("LongIntegerField", dbLong)
MyTable.Fields.Append MyTable.CreateField("MemoField", dbMemo)
MyTable.Fields.Append MyTable.CreateField("YesNoField", dbBoolean)
MyTable.Fields.Append MyTable.CreateField("DateTimeField", dbDate)
MyTable.Fields.Append MyTable.CreateField("IntegerField", dbInteger)
MyTable.Fields.Append MyTable.CreateField("SingleNumberField", dbSingle)
MyTable.Fields.Append MyTable.CreateField("DoubleNumberField", dbDouble)
CurrentDb.TableDefs.Append MyTable
Application.RefreshDatabaseWindow
End Sub

CurrentDb.TableDefs.Delete "NewTable"

CurrentDb.TableDefs("NewTable").Fields.Delete "TextField"

CurrentDb.TableDefs("NewTable").Fields("MemoField").DefaultValue = "abc"
CurrentDb.TableDefs("NewTable").Fields("MemoField").Required = True
CurrentDb.TableDefs("NewTable").Fields("MemoField").AllowZeroLength = False

CurrentDb.Execute "create index NewIndex on NewTable(TextField)"

CurrentDb.Execute ("drop index NewIndex on NewTable")

Sub CreateIndex()
Dim TDef As TableDef, Idx As Index
Dim db As Database

```

```

Set db = CurrentDb()
Set TDef = db.TableDefs("NewTable")
Set Idx = TDef.CreateIndex("MyIndex")
Idx.Fields.Append Idx.CreateField("LongIntegerField")
Idx.Primary = True
Idx.Required = True
Idx.Unique = True
TDef.Indexes.Append Idx
End Sub

CurrentDb.TableDefs("NewTable").Indexes.Delete "MyIndex"

```

Chapter 32

```

Sub Inventory()
Dim Obj As AccessObject, Dbs As Object, RecSet As Recordset
CurrentDb.Execute "delete * from tblObjectInventory"
Set RecSet = CurrentDb.OpenRecordset("tblObjectInventory")
Set Dbs = Application.CurrentData
For Each Obj In Dbs.AllTables
    RecSet.AddNew
    RecSet!ObjectName = Obj.Name
    RecSet!ObjectType = "Table"
    RecSet!DateCreated = Obj.DateCreated
    RecSet!DateModified = Obj.DateModified
    RecSet.Update
Next Obj
For Each Obj In Dbs.AllQueries
    RecSet.AddNew
    RecSet!ObjectName = Obj.Name
    RecSet!ObjectType = "Query"
    RecSet!DateCreated = Obj.DateCreated
    RecSet!DateModified = Obj.DateModified
    RecSet.Update
Next Obj
For Each Obj In CurrentProject.AllForms
    RecSet.AddNew
    RecSet!ObjectName = Obj.Name
    RecSet!ObjectType = "Form"
    RecSet!DateCreated = Obj.DateCreated
    RecSet!DateModified = Obj.DateModified
    RecSet.Update
Next Obj
For Each Obj In CurrentProject.AllReports
    RecSet.AddNew
    RecSet!ObjectName = Obj.Name
    RecSet!ObjectType = "Report"
    RecSet!DateCreated = Obj.DateCreated
    RecSet!DateModified = Obj.DateModified
    RecSet.Update
Next Obj
For Each Obj In CurrentProject.AllMacros
    RecSet.AddNew
    RecSet!ObjectName = Obj.Name
    RecSet!ObjectType = "Macro"
    RecSet!DateCreated = Obj.DateCreated
    RecSet!DateModified = Obj.DateModified
    RecSet.Update
Next Obj
For Each Obj In CurrentProject.AllModules
    RecSet.AddNew
    RecSet!ObjectName = Obj.Name
    RecSet!ObjectType = "Module"
    RecSet!DateCreated = Obj.DateCreated
    RecSet!DateModified = Obj.DateModified

```

```

        RecSet.Update
    Next Obj
    Set Dbs = Nothing
    Set Obj = Nothing
    Set RecSet = Nothing
End Sub

```

Chapter 33

```

Private Sub Form_Activate()
    Dim ch As Graph.Chart, Temp As String, PointCount As Integer
    Set ch = Me.Graph0.Object
    PointCount = ch.SeriesCollection(1).Points.Count
    For n = 1 To PointCount
        Select Case ch.SeriesCollection(1).Points(n).DataLabel.Text
            Case "USD"
                ch.SeriesCollection(1).Points(n).Interior.Color = QBColor(9)
            Case "GBP"
                ch.SeriesCollection(1).Points(n).Interior.Color = QBColor(10)
            Case "EUR"
                ch.SeriesCollection(1).Points(n).Interior.Color = QBColor(11)
            Case "AUD"
                ch.SeriesCollection(1).Points(n).Interior.Color = QBColor(12)
            Case "HKD"
                ch.SeriesCollection(1).Points(n).Interior.Color = QBColor(13)
        End Select
    Next n
    Set ch = Nothing
End Sub

```

Chapter 34

```

Private Type POINTAPI
    x As Long
    y As Long
End Type

Private Declare Function GetCursorPos Lib "user32" _
    (lpPoint As POINTAPI) As Long
Private Declare Function GetPixel Lib "gdi32" _
    (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
Private Declare Function GetWindowDC Lib "user32" (ByVal hwnd As Long) _
    As Long

Private Sub Graph0_MouseDown(Button As Integer, Shift As Integer, x As Single, _
    y As Single)
    Dim cPos As POINTAPI, Temp1 As Long, Temp2 As Long, Pointcount As Integer
    Dim ch As Graph.Chart, PCol As Long
    Call GetCursorPos(cPos)
    Temp = GetWindowDC(0)
    Temp1 = cPos.x
    Temp2 = cPos.y
    PCol = GetPixel(Temp, Temp1, Temp2)
    Set ch = Me.Graph0.Object
    Pointcount = ch.SeriesCollection(1).Points.Count
    For n = 1 To Pointcount
        If ch.SeriesCollection(1).Points(n).Interior.Color = PCol Then
            MsgBox ch.SeriesCollection(1).Points(n).DataLabel.Text
        End If
    Next n
End Sub

```

Chapter 35

```

Sub CreateSpreadsheet()
Dim RecSet As Recordset
Dim MyExcel As Excel.Application
Dim MyBook As Workbook
Dim MySheet As Worksheet
Set MyExcel = CreateObject("Excel.Application")
Set MyBook = MyExcel.Workbooks.Add
Set MySheet = MyBook.Worksheets(1)
Set RecSet = CurrentDb.OpenRecordset("customers")
MySheet.Range("a1").CopyFromRecordset RecSet
MyBook.SaveAs ("TestExcel")
MyBook.Close
MyExcel.Quit
Set RecSet = Nothing
Set MyExcel = Nothing
Set MyBook = Nothing
Set MySheet = Nothing
End Sub

```

```

Sub CreateSpreadsheet1()
Dim RecSet As Recordset
Dim MyExcel As Excel.Application
Dim MyBook As Workbook
Dim MySheet As Worksheet
Set MyExcel = CreateObject("Excel.Application")
Set MyBook = MyExcel.Workbooks.Open("MyRecSet", , True)
Set MySheet = MyBook.Worksheets(1)
Set RecSet = CurrentDb.OpenRecordset("customers")
MySheet.Range("a2").CopyFromRecordset RecSet
On Error Resume Next
Kill "TestExcel.xlsx"
On Error GoTo 0
MyBook.SaveAs ("TestExcel")
MyBook.Close
MyExcel.Quit
Set RecSet = Nothing
Set MyExcel = Nothing
Set MyBook = Nothing
Set MySheet = Nothing
End Sub

```

```

Sub CreateSpreadsheet2()
Dim RecSet As Recordset
Dim MyExcel As Excel.Application
Dim MyBook As Workbook
Dim MySheet As Worksheet
Dim strTemp As String
Dim Coun As Long
Dim Coun1 As Long
Set MyExcel = CreateObject("Excel.Application")
Set MyBook = MyExcel.Workbooks.Add
Set MySheet = MyBook.Worksheets(1)
strTemp = "SELECT Products.[Product Name], Sum([Unit  
Cost]*[quantity]) AS Cost "  
strTemp = strTemp & "FROM Products INNER JOIN [Purchase Order  
Details] "  
strTemp = strTemp & "ON Products.ID = [Purchase Order  
Details].[Product ID] "  
strTemp = strTemp & "GROUP BY Products.[Product Name];"  
Set RecSet = CurrentDb.OpenRecordset(strTemp)
Do Until RecSet.EOF
    Coun = Coun + 1
    MySheet.Range("a" & Coun).Value = RecSet![Product Name]
    MySheet.Range("b" & Coun).Value = RecSet!Cost
    Coun1 = Coun1 + 1
    If Coun1 = 5 Then
        Coun = Coun + 2
        Coun1 = 0
    End If
Loop

```

```

        End If
        RecSet.MoveNext
    Loop
    On Error Resume Next
    Kill "TestExcel.xlsx"
    On Error GoTo 0
    MyBook.SaveAs ("TestExcel")
    MyBook.Close
    MyExcel.Quit
    Set RecSet = Nothing
    Set MyExcel = Nothing
    Set MyBook = Nothing
    Set MySheet = Nothing
End Sub

Sub CreateSpreadsheet3()
    Dim RecSet As Recordset
    Dim MyExcel As Excel.Application
    Dim MyBook As Workbook
    Dim MySheet As Worksheet
    Dim strTemp As String
    Dim strCriteria As String
    Set MyExcel = CreateObject("Excel.Application")
    Set MyBook = MyExcel.Workbooks.Open("MyRecSet", , True)
    Set MySheet = MyBook.Worksheets(1)
    For n = 1 To 20
        For m = 1 To 20
            strCriteria = MySheet.Range(Chr(n + 64) & m).Value
            If Left(strCriteria, 1) = "!" Then
                On Error Resume Next
                strCriteria = Mid(strCriteria, 2)
                strTemp = "SELECT Products.[Product Name], Sum([Unit  
Cost]*[quantity]) AS Cost "  
strTemp = strTemp & "FROM Products INNER JOIN [Purchase Order  
Details]"  
strTemp = strTemp & "ON Products.ID = [Purchase Order  
Details].[Product ID] "  
strTemp = strTemp & "WHERE [Product Name]=' " & strCriteria & "' "  
strTemp = strTemp & "GROUP BY Products.[Product Name];"  
Set RecSet = CurrentDb.OpenRecordset(strTemp)
                If RecSet.RecordCount > 0 Then
                    MySheet.Range(Chr(n + 64) & m).Value = RecSet!cost
                Else
                    MySheet.Range(Chr(n + 64) & m).Value = 0
                End If
            End If
        Next m
    Next n
    On Error Resume Next
    Kill "TestExcel.xlsx"
    On Error GoTo 0
    MyBook.SaveAs ("TestExcel")
    MyBook.Close
    MyExcel.Quit
    Set RecSet = Nothing
    Set MyExcel = Nothing
    Set MyBook = Nothing
    Set MySheet = Nothing
End Sub

```

Chapter 36

```

Private Declare Function InternetOpen Lib "wininet.dll" Alias "InternetOpenA" _
    (ByVal sAgent As String, ByVal lAccessType As Long, ByVal sProxyName _
    As String, _
    ByVal sProxyBypass As String, ByVal lFlags As Long) As Long

```

```

Private Declare Function InternetConnect Lib "wininet.dll" _
Alias "InternetConnectA" _
(ByVal hInternetSession As Long, ByVal sServerName As String, _
ByVal nServerPort As Integer, ByVal sUsername As String, _
ByVal sPassword As String, ByVal lService As Long, _
ByVal lFlags As Long, ByVal lContext As Long) As Long

Private Declare Function FTPGETFile Lib "wininet.dll" Alias "FtpGetFileA" _
(ByVal hFtpSession As Long, ByVal lpszRemoteFile As String, _
ByVal lpszNewFile As String, ByVal fFailIfExists As Boolean, _
ByVal dwFlagsAndAttributes As Long, ByVal dwFlags As Long, _
ByVal dwContext As Long) As Boolean

Private Declare Function FtpPutFile Lib "wininet.dll" Alias "FtpPutFileA" _
(ByVal hFtpSession As Long, ByVal lpszLocalFile As String, _
ByVal lpszRemoteFile As String, ByVal dwFlags As Long, _
ByVal dwContext As Long) As Boolean

Private Declare Function InternetCloseHandle Lib "wininet.dll" _
(ByVal hInet As Long) As Integer

Sub GetFile()
Dim MyConn, MyINet, Chk As Boolean
Chk = False
MyINet = InternetOpen("MyFTP", 1, vbNullString, vbNullString, 0)
If MyINet > 0 Then
    MyConn = InternetConnect(MyINet, "MyOrg.MyServer.net", 21, "MyUserID", _
    "MyPassword", 1, 0, 0)
    If MyConn > 0 Then
        Chk = FTPGETFile(MyConn, "MyFolder/MyFileName.txt", _
        CurrentProject.Path & "\MyFileName.txt", 0, 0, 1, 0)
        InternetCloseHandle MyConn
    End If
    InternetCloseHandle MyINet
End If
If (Chk) Then
    MsgBox "File downloaded"
Else
    MsgBox "FTP Error"
End If
End Sub

Sub PutFile()
Dim MyConn, MyINet, Chk As Boolean
Chk = False
MyINet = InternetOpen("MyFTP", 1, vbNullString, vbNullString, 0)
If MyINet > 0 Then
    MyConn = InternetConnect(MyINet, "MyOrg.MyServer.net", 21, _
    "MyUserID", _ "MyPassword", 1, 0, 0)
    If MyConn > 0 Then
        Chk = FtpPutFile(MyConn, CurrentProject.Path & \test.txt", _
        " MyFolder/MyFileName.txt", 1, 0)
        InternetCloseHandle MyConn
    End If
    InternetCloseHandle MyINet
End If
If (Chk) Then
    MsgBox "File uploaded"
Else
    MsgBox "FTP Error"
End If
End Sub

```

Chapter 37

No code

Chapter 38

```
<customUI xmlns="http://schemas.microsoft.com/office/2009/07/customui" >
<ribbon startFromScratch="true">
<tabs>
    <tab id="customTab" label="MyMenu" >
        <group id="customGroup" label="My New Menu Structure">
            <box id="box1" boxStyle="horizontal">
                <menu id="Group1" label="Group1" >
                    <button id="id1" label="Option1" onAction="Action1"/>
                    <menuSeparator id="sep1" />
                    <button id="id2" label="Option2" onAction="Action2"/>
                    <button id="id3" label="Option3" onAction="Action3"/>
                </menu>

                <button id="SP1" label=" " enabled="false"/>
                <menu id="Group2" label="Group2" >
                    <button id="id4" label="Option4" onAction="Action4"/>
                    <button id="id5" label="Option5" onAction="Action5"/>
                    <menuSeparator id="sep2" />
                    <menu id="Group3" label="Group3" >
                        <button id="id7" label="Option7" onAction="Action7"/>
                        <button id="id8" label="Option8" onAction="Action8"/>
                    </menu>

                    <button id="id6" label="Option6" onAction="Action6"/>
                </menu>
            </box>
        </group>
    </tab>
</tabs>
</ribbon>
</customUI>

'Callback for id1 onAction
Sub Action1(control as IRibbonControl)
End Sub

'Callback for id2 onAction
Sub Action2(control as IRibbonControl)
End Sub

'Callback for id3 onAction
Sub Action3(control as IRibbonControl)
End Sub
'Callback for id4 onAction
Sub Action4(control as IRibbonControl)
End Sub
'Callback for id5 onAction
Sub Action5(control as IRibbonControl)
End Sub
'Callback for id6 onAction
Sub Action6(control as IRibbonControl)
End Sub

'Callback for id1 onAction
Sub Action1(control as IRibbonControl)
MsgBox "You selected Option1"
End Sub
```

Chapter 39

```
SELECT DISTINCT SupplierSize FROM tblSupplier;
```

```

Private Sub Combo0_Change()
Dim strTemp As String
Me.Combo0.SetFocus
strTemp = Me.Combo0.text
Me.Combo2.SetFocus
Me.Combo2.RowSource = _
"select distinct SupplierName from tblSupplier where SupplierSize='" _
& strTemp & "'"
If strTemp = "Small" Then
    Me.Text4.Visible = False
Else
    Me.Text4.Visible = True
End If
Me.Combo0.SetFocus
End Sub

Private Sub Form_Current()
Combo0_Change
End Sub

```

Chapter 40

```

Public Declare Function GetUserName Lib "advapi32.dll" Alias _
"GetUserNameA" (ByVal lpBuffer As String, nSize As Long) As Long

Public Function ReturnUserLevel()
Dim strUser As String, X As Integer, RecSet as Recordset
strUser = Space$(256)
X = GetUserName(strUser, 256)
strUser = RTrim(strUser)
strUser = Left(strUser, Len(strUser) - 1)
Set RecSet = CurrentDb.OpenRecordset _
("select * tblUserRights where UserName='" & strUser & "'")
If RecSet.RecordCount Then
    ReturnUserLevel = RecSet!UserLevel
Else
    ReturnUserLevel = "unknown"
End If
Set RecSet = Nothing
End Function

If ReturnUserLevel <> "administrator" Then
MsgBox "You are not authorized to view this screen", vbCritical
DoCmd.Close
End If

If ReturnUserLevel = "unknown" then DoCmd.Close
If ReturnUserLevel = "basic" Then
    Me.AllowAdditions = False
    Me.AllowEdits = False
    Me.AllowDeletions = False
Else
    Me.AllowAdditions = True
    Me.AllowEdits = True
    Me.AllowDeletions = True
End If

If ReturnUserLevel = "unknown" then DoCmd.Close
If ReturnUserLevel = "basic" Or ReturnUserLevel = "novice" Then
    Me.Text1.Visible = False
    Me.Label2.Visible = False
Else
    Me.Text1.Visible = True
    Me.Label2.Visible = True
End If

```

