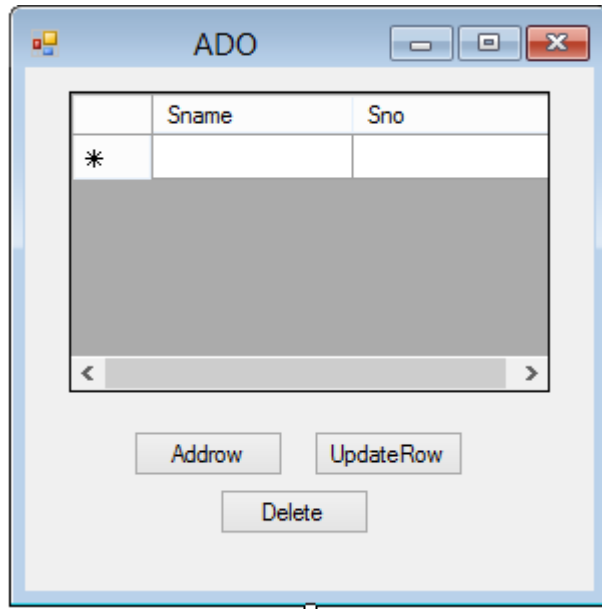


ADO

ADO WIZARD



```
Imports System.Data.OleDb
```

```
Public Class Form1
```

```
    Dim sno As Integer  
    Dim sname As String  
    Dim rw As DataRow  
    Dim rno As Integer  
    Dim i, rwno As Integer  
    Dim colname As String  
    Dim value As Object
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load  
        'TODO: This line of code loads data into the  
'TestDbDataSet.Student' table. You can move, or remove it, as needed.  
        Me.StudentTableAdapter.Fill(Me.TestDbDataSet.Student)
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnAddrow.Click  
        rw = TestDbDataSet.Student.NewRow  
        sno = InputBox("Enter the Roll no of the Student:")  
        sname = InputBox("Enter the Name of the Student:")  
        rw.Item("Sno") = sno  
        rw.Item("Sname") = sname  
        Try  
            TestDbDataSet.Student.Rows.Add(rw)  
            i = StudentTableAdapter.Update(TestDbDataSet)  
        Catch ex As Exception  
            MessageBox.Show(ex.Message)  
        End Try  
        'Displays number of rows updated.
```

```

        MessageBox.Show("no of rows updated=" & i)
    End Sub

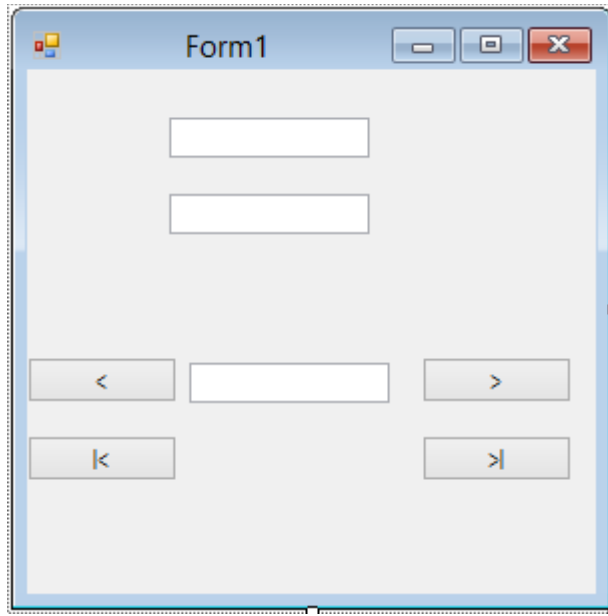
    Private Sub btnUpdateRow_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnUpdateRow.Click
        colname = InputBox("Enter the name of the Column to be updated")
        rwno = InputBox("Enter the Row Number to be updated: Row No starts
from 0")
        value = InputBox("Enter the value to be entered into the Student
table")
        Try
            'Update the column in the Student table.
            TestDbDataSet.Student.Rows(rwno).Item(colname) = value
            'Update the Student table in the testdb database.
            i = StudentTableAdapter.Update(TestDbDataSet)
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
        'Displays number of rows updated.
        MessageBox.Show("no of rows updated=" & i)
    End Sub

    Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
        rno = InputBox("Enter the Row no to be deleted: Row no starts from
0")
        Try
            'Delete a row from the Student table.
            TestDbDataSet.Student.Rows(rno).Delete()
            'Update the Student table in the testdb database.
            i = StudentTableAdapter.Update(TestDbDataSet)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        'Displays number of rows updated.
        MessageBox.Show("no of rows updated=" & i)
    End Sub

End Class

```

ADO Navigation (Coding Method)



```
Imports System.Data
Imports System.Data.OleDb
Public Class Form1
    Dim connectionString As String
    Dim objconnection As New OleDbConnection
    Dim objDataSet As New DataSet()
    Dim objDataAdapter As New OleDbDataAdapter
    Dim objDataView As DataView
    Dim objCurrencyManager As CurrencyManager
    Dim querystr As String
    Dim cmd As New OleDbCommand
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Acheiving Connection
        connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\admin\Desktop\student.accdb"
        objconnection = New OleDbConnection(connectionString)
        Try
            objconnection.Open()
            MsgBox("Connection Open ! ")
            objconnection.Close()
        Catch ex As Exception
            MsgBox("Can not open connection ! ")
        End Try

        'Loading Adapter
        querystr = "SELECT * from stu;"
        cmd = New OleDbCommand(querystr, objconnection)
        objDataAdapter.SelectCommand = cmd

        'Fill adapter
        objDataAdapter.Fill(objDataSet, "stu")
    End Sub
End Class
```

```

        'Filling View and currency
        objDataView = New DataView(objDataSet.Tables("stu"))
        objCurrencyManager = CType(Me.BindingContext(objDataView),
CurrencyManager)

        'Binding controls
        ' Clear any previous bindings..
        txtName.DataBindings.Clear()
        txtNo.DataBindings.Clear()

        ' Add new bindings to the DataView object..
        txtName.DataBindings.Add("Text", objDataView, "Sname")
        txtNo.DataBindings.Add("Text", objDataView, "Sno")

        'Show Position
        ShowPosition()
    End Sub
    Private Sub ShowPosition()
        ' Display the current position and the number of records
        txtRecordPosition.Text = objCurrencyManager.Position + 1 & " of " &
objCurrencyManager.Count()
    End Sub

    Private Sub btnMoveFirst_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnMoveFirst.Click
        ' Set the record position to the first record..
        objCurrencyManager.Position = 0
        ' Show the current record position..
        ShowPosition()
    End Sub

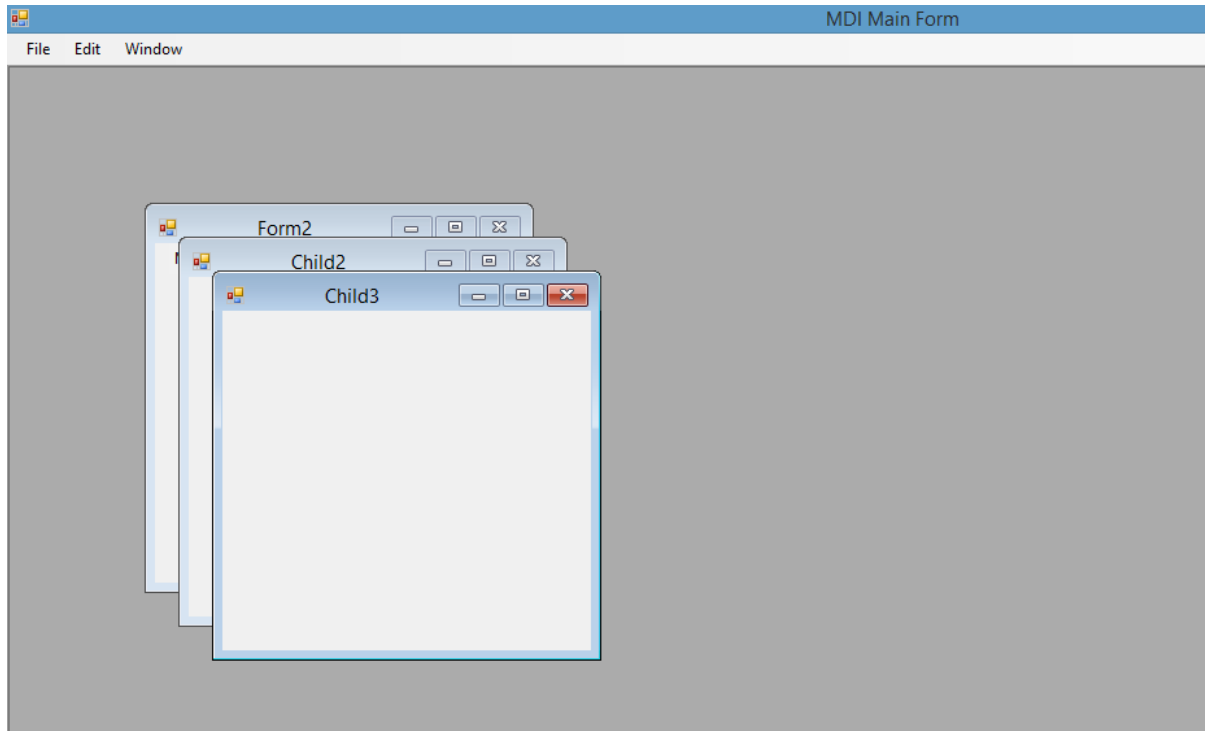
    Private Sub btnPrev_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrev.Click
        ' Move to the previous record..
        objCurrencyManager.Position -= 1
        ' Show the current record position..
        ShowPosition()
    End Sub

    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNext.Click
        ' Move to the next record..
        objCurrencyManager.Position += 1
        ' Show the current record position..
        ShowPosition()
    End Sub

    Private Sub btnMoveLast_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMoveLast.Click
        ' Set the record position to the last record..
        objCurrencyManager.Position = objCurrencyManager.Count - 1
        ' Show the current record position..
        ShowPosition()
    End Sub
End Class

```

MDI



```
Public Class frmMain
    Private Sub mnuFChild_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFChild.Click
        Dim frm As New frmChildwithMenus()
        frm.MdiParent = Me
        frm.Show()
    End Sub

    Private Sub mnuCascade_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuCascade.Click
        Me.LayoutMdi (MdiLayout.Cascade)
    End Sub

    Private Sub mnuTileHorizontal_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuTileHorizontal.Click
        Me.LayoutMdi (MdiLayout.TileHorizontal)
    End Sub

    Private Sub mnuTileVertical_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles mnuTileVertical.Click
        Me.LayoutMdi (MdiLayout.TileVertical)
    End Sub
End Class
```

```

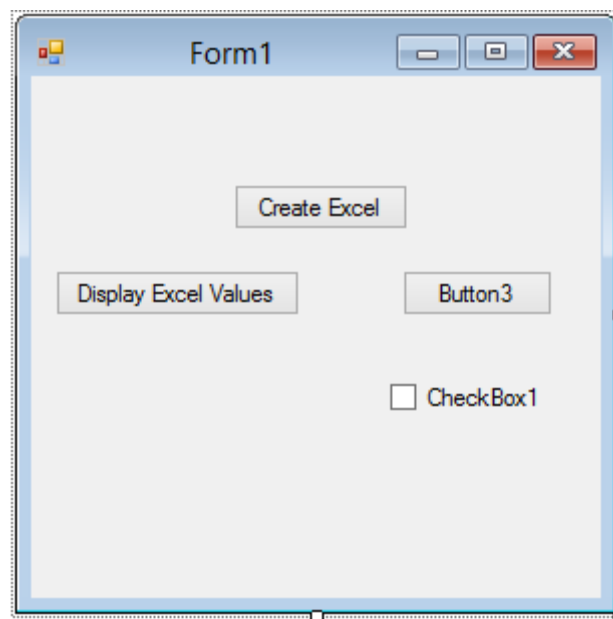
        Private Sub Child2ToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Child2ToolStripMenuItem.Click
            Dim frm As New Child2
            frm.MdiParent = Me
            frm.Show()
        End Sub

        Private Sub Child3ToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Child3ToolStripMenuItem.Click
            Dim frm As New Child3
            frm.MdiParent = Me
            frm.Show()
        End Sub

End Class

```

Automation with Excel (COM)



```

Imports Microsoft.Office.Interop
Public Class Form1
    Dim objApp As Excel.Application
    Dim objBook As Excel._Workbook

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim objBooks As Excel.Workbooks
        Dim objSheets As Excel.Sheets
        Dim objSheet As Excel._Worksheet
        Dim range As Excel.Range
    End Sub

```

```

' Create a new instance of Excel and start a new workbook.
objApp = New Excel.Application()
objBooks = objApp.Workbooks
objBook = objBooks.Add
objSheets = objBook.Worksheets
objSheet = objSheets(1)

'Get the range where the starting cell has the address
'm_sStartingCell and its dimensions are m_iNumRows x m_iNumCols.
range = objSheet.Range("A1", Reflection.Missing.Value)
range = range.Resize(5, 5)

If (Me.FillWithStrings.Checked = False) Then
    'Create an array.
    Dim saRet(5, 5) As Double

    'Fill the array.
    Dim iRow As Long
    Dim iCol As Long
    For iRow = 0 To 5
        For iCol = 0 To 5

            'Put a counter in the cell.
            saRet(iRow, iCol) = iRow * iCol
        Next iCol
    Next iRow

    'Set the range value to the array.
    range.Value = saRet
Else
    'Create an array.
    Dim saRet(5, 5) As String

    'Fill the array.
    Dim iRow As Long
    Dim iCol As Long
    For iRow = 0 To 5
        For iCol = 0 To 5

            'Put the row and column address in the cell.
            saRet(iRow, iCol) = iRow.ToString() + "|" +
iCol.ToString()
        Next iCol
    Next iRow

    'Set the range value to the array.
    range.Value = saRet
End If

'Return control of Excel to the user.
objApp.Visible = True
objApp.UserControl = True

'Clean up a little.
range = Nothing
objSheet = Nothing
objSheets = Nothing
objBooks = Nothing

End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim objSheets As Excel.Sheets
    Dim objSheet As Excel._Worksheet
    Dim range As Excel.Range

    'Get a reference to the first sheet of the workbook.
    On Error GoTo ExcelNotRunning
    objSheets = objBook.Worksheets
    objSheet = objSheets(1)

ExcelNotRunning:
    If (Not (Err.Number = 0)) Then
        MsgBox.Show("Cannot find the Excel workbook. Try clicking
Button1 to " + _
            "create an Excel workbook with data before running Button2.", _
            "Missing Workbook?")

        'We cannot automate Excel if we cannot find the data we
created,
        'so leave the subroutine.
        Exit Sub
    End If

    'Get a range of data.
    range = objSheet.Range("A1", "E5")

    'Retrieve the data from the range.
    Dim saRet(,) As Object
    saRet = range.Value

    'Determine the dimensions of the array.
    Dim iRows As Long
    Dim iCols As Long
    iRows = saRet.GetUpperBound(0)
    iCols = saRet.GetUpperBound(1)

    'Build a string that contains the data of the array.
    Dim valueString As String
    valueString = "Array Data" + vbCrLf

    Dim rowCounter As Long
    Dim colCounter As Long
    For rowCounter = 1 To iRows
        For colCounter = 1 To iCols

            'Write the next value into the string.
            valueString = String.Concat(valueString, _
                saRet(rowCounter, colCounter).ToString() + ", ")

        Next colCounter

        'Write in a new line.
        valueString = String.Concat(valueString, vbCrLf)
    Next rowCounter

    'Report the value of the array.
    MsgBox.Show(valueString, "Array Values")

    'Clean up a little.

```



```

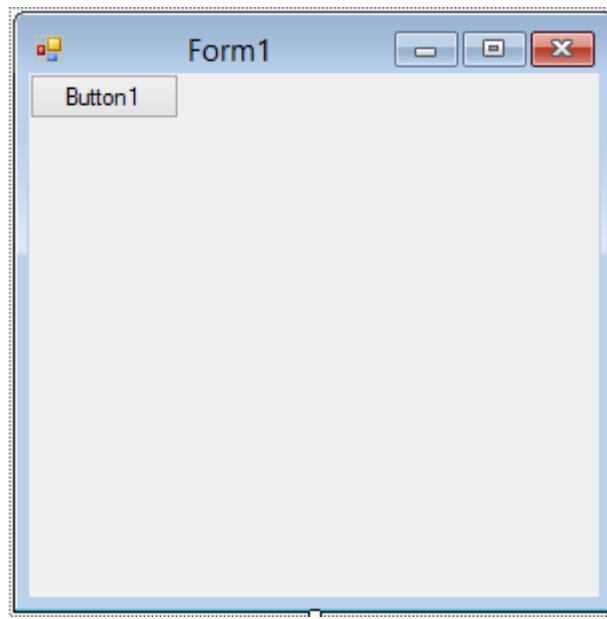
range = Nothing
objSheet = Nothing
objSheets = Nothing

```

```
End Sub
```

```
End Class
```

Automation with Word (COM)



```

Imports Microsoft.Office.Interop
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim oWord As Word.Application
        Dim oDoc As Word.Document
        Dim oTable As Word.Table
        Dim oPara1 As Word.Paragraph, oPara2 As Word.Paragraph
        Dim oPara3 As Word.Paragraph, oPara4 As Word.Paragraph
        Dim oRng As Word.Range
        Dim oShape As Word.InlineShape
        Dim oChart As Object
        Dim Pos As Double

        'Start Word and open the document template.
        oWord = CreateObject("Word.Application")
        oWord.Visible = True
        oDoc = oWord.Documents.Add

        'Insert a paragraph at the beginning of the document.
        oPara1 = oDoc.Content.Paragraphs.Add
        oPara1.Range.Text = "Heading 1"
    
```

```

oPara1.Range.Font.Bold = True
oPara1.Format.SpaceAfter = 24      '24 pt spacing after paragraph.
oPara1.Range.InsertParagraphAfter()

'Insert a paragraph at the end of the document.
'** \endofdoc is a predefined bookmark.
oPara2 =
oDoc.Content.Paragraphs.Add(oDoc.Bookmarks.Item("\endofdoc").Range)
oPara2.Range.Text = "Heading 2"
oPara2.Format.SpaceAfter = 6
oPara2.Range.InsertParagraphAfter()

'Insert another paragraph.
oPara3 =
oDoc.Content.Paragraphs.Add(oDoc.Bookmarks.Item("\endofdoc").Range)
oPara3.Range.Text = "This is a sentence of normal text. Now here is
a table:"
oPara3.Range.Font.Bold = False
oPara3.Format.SpaceAfter = 24
oPara3.Range.InsertParagraphAfter()

'Insert a 3 x 5 table, fill it with data, and make the first row
'bold and italic.
Dim r As Integer, c As Integer
oTable = oDoc.Tables.Add(oDoc.Bookmarks.Item("\endofdoc").Range, 3,
5)
oTable.Range.ParagraphFormat.SpaceAfter = 6
For r = 1 To 3
    For c = 1 To 5
        oTable.Cell(r, c).Range.Text = "r" & r & "c" & c
    Next
Next
oTable.Rows.Item(1).Range.Font.Bold = True
oTable.Rows.Item(1).Range.Font.Italic = True

'Add some text after the table.
'oTable.Range.InsertParagraphAfter()
oPara4 =
oDoc.Content.Paragraphs.Add(oDoc.Bookmarks.Item("\endofdoc").Range)
oPara4.Range.InsertParagraphBefore()
oPara4.Range.Text = "And here's another table:"
oPara4.Format.SpaceAfter = 24
oPara4.Range.InsertParagraphAfter()

'Insert a 5 x 2 table, fill it with data, and change the column
widths.
oTable = oDoc.Tables.Add(oDoc.Bookmarks.Item("\endofdoc").Range, 5,
2)
oTable.Range.ParagraphFormat.SpaceAfter = 6
For r = 1 To 5
    For c = 1 To 2
        oTable.Cell(r, c).Range.Text = "r" & r & "c" & c
    Next
Next
oTable.Columns.Item(1).Width = oWord.InchesToPoints(2)      'Change
width of columns 1 & 2
oTable.Columns.Item(2).Width = oWord.InchesToPoints(3)

'Keep inserting text. When you get to 7 inches from top of the
'document, insert a hard page break.
Pos = oWord.InchesToPoints(7)

```

```

oDoc.Bookmarks.Item("\endofdoc").Range.InsertParagraphAfter()
Do
    oRng = oDoc.Bookmarks.Item("\endofdoc").Range
    oRng.ParagraphFormat.SpaceAfter = 6
    oRng.InsertAfter("A line of text")
    oRng.InsertParagraphAfter()
Loop While Pos >=
oRng.Information(Word.WdInformation.wdVerticalPositionRelativeToPage)
oRng.Collapse(Word.WdCollapseDirection.wdCollapseEnd)
oRng.InsertBreak(Word.WdBreakType.wdPageBreak)
oRng.Collapse(Word.WdCollapseDirection.wdCollapseEnd)
oRng.InsertAfter("We're now on page 2. Here's my chart:")
oRng.InsertParagraphAfter()

'Insert a chart and change the chart.
oShape =
oDoc.Bookmarks.Item("\endofdoc").Range.InlineShapes.AddOLEObject( _
    ClassType:="MSGraph.Chart.8", FileName _
    :="", LinkToFile:=False, DisplayAsIcon:=False)
oChart = oShape.OLEFormat.Object
oChart.charttype = 4 'xlLine = 4
oChart.Application.Update()
oChart.Application.Quit()
'If desired, you can proceed from here using the Microsoft Graph
'Object model on the oChart object to make additional changes to
the
'chart.
oShape.Width = oWord.InchesToPoints(6.25)
oShape.Height = oWord.InchesToPoints(3.57)

'Add text after the chart.
oRng = oDoc.Bookmarks.Item("\endofdoc").Range
oRng.InsertParagraphAfter()
oRng.InsertAfter("THE END.")

'All done. Close this form.
Me.Close()

End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub
End Class

```