

Appendix A

This appendix contains the Visual Basic code for the implementation of the Mechanical Desktop System. Microsoft® Visual Basic 6.0, installed on a Pentium II (200Mhz) computer running Windows NT 4.0, is the programming language that has been used in this research to develop the graphical user interface. TKSolver® was used as an implicit solving tool, Microsoft® Access 97 was used as database to store the design equations, tables and standards, and Microsoft® Word 97 to document the design.

frmMain.frm

This is the source code for the Main form.

Option Explicit

'--- Variable to store the equation number

Dim k As String

'--- Variable to store the element flag

Dim var1 As String

'--- Variable for storing the results of a calculation

Dim retva1 As String

'--- Variable for passing values to TK Solver

Dim retv2 As String

'--- Stores the filename for tkw model depending upon equation

Dim modelname As String

'--- Variable checks for the equation

Dim eqset As Boolean

'--- Variable holding the value of number of result boxes to display

Dim NOB As Integer

This function is executed on clicking the Close Button on the main form

Private Sub CmdClose_Click()

 ' ---Closes the form

 End

End Sub

This function is executed during the lost focus event of the combo box containing equations

```
Private Sub Combo2_LostFocus()
```

```
    '--- Variable to store the number of labels
```

```
    Dim i As Integer
```

```
    '--- Variable to store the query
```

```
    Dim l as Integer
```

```
    '--- Variable to store the query
```

```
    Dim str As String
```

```
    '--- Open a new record set
```

```
    Dim rs As New ADODB.Recordset
```

```
    '--- Query to select the equation number based on equation
```

```
    str = "select * from EquationTable where equation = " & Combo2.Text & " "
```

```
    rs.Open str, connect
```

```
    '--- Based on equation number assign the caption to the labels, display equation
```

```
    '--- number and specify the TK Solver model to open
```

```
    Select Case rs("eq_number")
```

```
        Case 17.1
```

```
            lblPrompt.Caption = "Select the value of Ks from Table 17-9"
```

```
            lblEqNos.Caption = "17.1"
```

```
            lblPageNo.Caption = " Page 770"
```

```
            modelname = "C:\Machine_Elements\Belt\Eq_17_1.tkw"
```

```
        Case 17.2
```

```
            lblPrompt.Caption = "Please select initial sheave diameters from Table 17-2"
```

```
            lblEqNos.Caption = "17.2"
```

```
            lblPageNo.Caption = " Page 770"
```

```
modelname = "C:\Machine_Elements\Belt\Eq_17_2.tkw"
```

Case 17.3

```
lblPrompt.Caption = ""
```

```
lblEqNos.Caption = "17.3"
```

```
lblPageNo.Caption = " Page 770"
```

```
modelname = "C:\Machine_Elements\Belt\Eq_17_3.tkw"
```

Case 17.4

```
lblPrompt.Caption = ""
```

```
lblEqNos.Caption = "17.4"
```

```
lblPageNo.Caption = " Page 765"
```

```
modelname = "C:\Machine_Elements\Belt\Eq_17_4.tkw"
```

Case 17.5

```
lblPrompt.Caption = "Select value of x from Table 17-4 and standardize the belt_  
from Table 17-7 "
```

```
lblEqNos.Caption = "17.5"
```

```
lblPageNo.Caption = " Page 771"
```

```
modelname = "C:\Machine_Elements\Belt\Eq_17_5.tkw"
```

Case 17.6

```
lblPrompt.Caption = "Select value of Ka from Table 17-6 based on D/d ratio and  
C1,C2,C3 and C4 values from Table 17-5"
```

```
lblEqNos.Caption = "17.6"
```

```
lblPageNo.Caption = "Page 766"
```

```
modelname = "C:\Machine_Elements\Belt\Eq_17_6.tkw"
```

Case 17.7

```
lblPrompt.Caption = "Select values of K1 and K2 from Graph 17-4, based on  
angle of contact and Table 17-7 based on belt length, respectively"
```

```
lblEqNos.Caption = "17.7"
```

```
lblPageNo.Caption = "Page 768"
```

```
modelname = "C:\Machine_Elements\Belt\Eq_17_7.tkw"
```

Case 17.8

```
lblEqNos.Caption = "17.8"
```

lblPageNo.Caption = "Page 771 "
modelName = "C:\Machine_Elements\Belt\Eq_17_8.tkw"

Case 17.9

lblEqNos.Caption = "17.9"
lblPageNo.Caption = "Page 771 "
modelName = "C:\Machine_Elements\Belt\Eq_17_9.tkw"

Case 18.1

lblPrompt.Caption = "Select service factor Ks from Table 17-14"
lblEqNos.Caption = "18.1"
lblPageNo.Caption = "Page 779 "
modelName = "C:\Machine_Elements\Chain&Sprocket\Eq_18_1.tkw"

Case 18.2

lblPrompt.Caption = "Select Hrc value from Table 17-11,K1 from Table 17-12
and K2 from Table 17-13"
lblEqNos.Caption = "18.2"
lblPageNo.Caption = "Page 780 "
modelName = "C:\Machine_Elements\Chain&Sprocket\Eq_18_2.tkw"

Case 18.3

lblPrompt.Caption = "Select the value of pitch p for the chain chosen from Table
17 -10"
lblEqNos.Caption = "18.3"
lblPageNo.Caption = "Page 780 "
modelName = "C:\Machine_Elements\Chain&Sprocket\Eq_18_3.tkw"

Case 18.4

lblEqNos.Caption = "18.4"
lblPageNo.Caption = "Page 780 "
modelName = "C:\Machine_Elements\Chain&Sprocket\Eq_18_4.tkw"

Case 18.5

lblEqNos.Caption = "18.5"
lblPageNo.Caption = "Page 780 "
modelName = "C:\Machine_Elements\Chain&Sprocket\Eq_18_5.tkw"

```

End Select
'--- Check whether the next equation has been selected and unload the text boxes
    If eqset = True Then
        SendtoSolver_Click
        eqset = False
    Else
        For i = NOB - 1 To 1 Step -1
            l = Label2(i)
            Unload TxtResult(i)
            Unload Label2(i)
        Next i
        '--- Closes the connection to previous TK Solver model
        tkw.hidewindow
        '--- Send parameter values to TK Solver
        SendtoSolver_Click
    End If

```

End Sub

This function sends the equations and parameter values to TK Solver and back to the interface.

```
Private Sub SendtoSolver_Click()
```

```
'--- Intermediate variables used for sending parameters to TK Solver
```

```
Dim i, val, f, g As Integer
```

```
'--- This variable stores the value of the calculated parameter
```

```
Dim vn As String
```

```

'--- These variables store the return variable status
Dim ret, ret1 As Integer
'--- This variable stores the number of variables in an equation
Dim noVar As Integer
'--- Open connection to the TK Model for the particular equation
    Set tkw = GetObject(modelname, "TKW.Document")
'--- Send the rule to the particular model
    ret = tkw.setrule(1, "r", Combo2.Text)
'--- Store the number of parameters in the equation
    noVar = tkw.getlastline("v")
'--- Initially make the dynamic textboxes and labels invisible
    TxtResult(0).Visible = False
    Label2(0).Visible = False
'--- Create the dynamic textboxes and labels based on the number of parameters
    For i = 1 To noVar
        Load TxtResult(i)
        Load Label2(i)
        TxtResult(i).Left = TxtResult(i - 1).Left + TxtResult(i - 1).Width + 100
        Label2(i).Left = Label2(i - 1).Left + Label2(i - 1).Width + 100
        TxtResult(i).Visible = True
        Label2(i).Visible = True
        TxtResult(i) = ""
        Label2(i) = tkw.getsheetcell("v", i, 3)
    Next i
    NOB = i
'--- The parameter values finalized earlier are reassigned
Select Case k
    Case 17.3
        TxtResult(3).Text = BeltNewD1
        TxtResult(2).Text = BeltNewD2
    Case 17.4

```

```

    TxtResult(3).Text = BeltNewD2
    TxtResult(4).Text = BeltNewD1
    TxtResult(2).Text = BeltCen
Case 17.5
    TxtResult(2).Text = BeltLp
Case 17.6
    Dbyd1.Visible = True
    Text14.Visible = True
    Text14.Text = BeltNewD2 / BeltNewD1
    TxtResult(4) = BeltNewD1
Case 17.7
    TxtResult(4) = BeltHr
    Text14.Visible = False
    Dbyd1.Visible = False
Case 17.8
    TxtResult(2).Text = BeltNewD1
    TxtResult(3).Text = Text1.Text
    Text14.Visible = False
    Dbyd1.Visible = False
Case 17.9
    TxtResult(2).Text = BeltPd
    TxtResult(3).Text = BeltH_r
    Text14.Visible = False
    Dbyd1.Visible = False
End Select
'--- Save the model for the current equation
tkw.savemodel

End Sub

```

This function is executed on click event of the Calculate command button

```
Private Sub cmdCalculate_Click()
```

```
    Dim i, val, f, g As Integer
```

```
    Dim retva, retV, vn, v As String
```

```
    Dim ret As Integer
```

```
    '--- Check whether a number has been entered in the text boxes
```

```
    f = tkw.getlastline("v")
```

```
    For i = 1 To f
```

```
        vn = Trim(Label2(i).Caption)
```

```
        If (TxtResult(i).Text > Chr(65) And TxtResult(i).Text < Chr(90)) Or
```

```
            (TxtResult(i).Text > Chr(97) And TxtResult(i).Text < Chr(122)) Then
```

```
            MsgBox " Enter A Number"
```

```
            Exit Sub
```

```
            TxtResult(i).SetFocus
```

```
        Else
```

```
            '--- Assign the value of the textbox to the variable
```

```
            val = TxtResult(i).Text
```

```
            '--- Set the value in the variable sheet of TK Solver as input
```

```
            ret = tkw.setvalue(vn, "i", val)
```

```
        End If
```

```
    Next i
```

```
    '--- Solve command to TK Solver
```

```
    tkw.solve
```

```
    '--- Get the last line from TK Solver sheet
```

```
    f = tkw.getlastline("v")
```

```
    '--- Get the values from the variable sheet
```

```
    For i = 1 To f
```

```

retV = tkw.getsheetcell("v", i, 4)
retva = Trim(tkw.getsheetcell("v", i, 3))
'--- Add the variable name and value to the listbox
If Trim(retV) <> "" Then
    List1.AddItem retva & " " & retV
    retva1 = retva
    retv2 = retV
End If
'--- If the return value is not null, then assign the value
If retV <> "" Then
    v = retV
End If
Next i
'--- The finalized variable values are assigned to a temporary variable
Select Case k
Case 17.1
    BeltPd = v
Case 17.2
    BeltNewD1 = TxtResult(3)
    BeltNewD2 = TxtResult(4)
Case 17.3
    BeltCen = TxtResult(4)
Case 17.4
    BeltLp = v
Case 17.5
    Text11.Text = v
Case 17.6
    BeltHr = v
Case 17.7
    BeltH_r = v
End Select

```

End Sub

Initializations performed on loading the form

Private Sub Form_Load()

'---Set the connection

Set connect = New ADODB.Connection

'---Connect to the database with given system DSN

connect.Open "DSN=Model"

'---Set a new recordset

Set rs = New ADODB.Recordset

'---Load the sections from the table

sql = "select section from Table17_2"

'---Connect to the database and load the equations

rs.Open sql, connect

Combo1.Clear

rs.MoveFirst

Do While Not rs.EOF

 Combo1.AddItem (rs("section"))

 rs.MoveNext

Loop

rs.Close

eqset = True

'---Initialize the menu to show no tables and graphs

mnuTableBelt.Visible = False

mnuGraphs.Visible = False

End Sub

This function is executed on clicking the belt menu item

```
Private Sub mnuBelt_Click()
```

```
    '--- Declare a new recordset
```

```
    Set rs = New ADODB.Recordset
```

```
    '--- Select all values from the table ordering by equation number
```

```
    sql = "select * from EquationTable order by eq_number"
```

```
    '--- Connect to the database
```

```
    rs.Open sql, connect
```

```
    '--- Clear the combobox
```

```
    Combo2.Clear
```

```
    '--- Move to the first record
```

```
    rs.MoveFirst
```

```
    '---Assign the flag value
```

```
    var1 = rs("flag")
```

```
    '---Load the equations for belt into the combobox
```

```
    Do While Not rs.EOF
```

```
        If rs("flag") = "B" Then
```

```
            Combo2.AddItem (rs("Equation"))
```

```
        End If
```

```
        rs.MoveNext
```

```
    Loop
```

```
    '---Load the Tables and Graphs for Belts
```

```
    mnuTableChain.Visible = False
```

```
    mnuTableBelt.Visible = True
```

```
    mnuGraphs.Visible = True
```

```
    lblType.Visible = False
```

```
    '--- Set the caption and show the form for entering specifications
```

```
    Form3.Caption = "Please enter the specifications/limitations imposed by the design"
```

```

Form3.Show
'--- Set the focus to the first textbox
Form3.TxtSpec(1).SetFocus
'--- Make the prompt visible
lblPrompt.Visible = True

```

End Sub

```

*****

```

This function is executed when clicking the Chain and Sprocket menu item

```

*****

```

```

Private Sub mnuChainSprocket_Click()

```

```

'---Declare a new database recordset
Set rs = New ADODB.Recordset
'---Select all values from the table ordering by equation number
sql = "select * from EquationTable order by eq_number"
'---Connect to the database
rs.Open sql, connect
'---Clear the combobox
Combo2.Clear
'---Move to the first record
rs.MoveFirst
'---Assign the flag value
var1 = rs("flag")
'---Load the equations for Chain and Sprocket into the combobox
Do While Not rs.EOF
If rs("flag") = "C" Then
Combo2.AddItem (rs("Equation"))
End If

```

```

        rs.MoveNext
    Loop
    '---Set the caption and show the form for entering specifications
    Form3.Caption = "Please enter the specifications/limitations imposed by the design"
    Form3.Show
    '---Set the focus to the first textbox
    Form3.TxtSpec(1).SetFocus
    '---Load the tables for Chain and sprocket and make that for the belt invisible
    mnuTableBelt.Visible = False
    mnuTableChain.Visible = True
    mnuGraphs.Visible = False
    lblPrompt.Caption = ""

End Sub

```

```

*****
This function is executed on clicking the Display Information menu item
*****

```

```

Private Sub mnuDisplayInfo_Click()

    '--- Show the form to display information
    frmFile.Show

End Sub

```

```

*****
This function is executed on clicking the Figure 17_4 menu item
*****

```

```

Private Sub mnuF_17_4_Click()

```

```
'---Load the graph and show the form
Form4.Image1.Picture = LoadPicture("C:\Anuj_thesis\Figure17_4.jpg")
Form4.Caption = "Correction Factor K1 for angle of contact upto 180 degrees"
Form4.Show
```

End Sub

```
*****
```

This function is executed on clicking the Input Information menu item

```
*****
```

```
Private Sub mnuInputInfo_Click()
```

```
'--- Show the form to input information
```

```
frmInputInfo.Show
```

End Sub

```
*****
```

This function is executed on clicking the TK Solver menu item

```
*****
```

```
Private Sub mnuopenTK_Click()
```

```
'--- External call to initiate TK Solver
```

```
Shell ("C:\Program Files\UTS\TK Solver 3.32\TKW32")
```

End Sub

This function is executed on clicking the open URL menu item

```
Private Sub MnuOpenURL_Click()
```

```
    '--- Open a new Browser form
    Dim frmB As New frmBrowser
    '--- Setting the default starting address
    frmB.StartingAddress = "http://www.google.com"
    '--- Show the form
    frmB.Show
```

```
End Sub
```

This function is executed on clicking the Table 17-2 menu item and displays the contents of the table

```
Private Sub mnuT17_2_Click()
```

```
    '--- Variable declaration for sql query and for accessing table data
    Dim str As String
    Dim rs As New ADODB.Recordset
    '--- Assign the column headers for the ListBox
    Dim itmX As ListItem
    Dim clmX As ColumnHeader
    Set clmX = Form2.ListView1.ColumnHeaders. _
    Add(, , "Type", Form2.ListView1.Width / 4)
    Set clmX = Form2.ListView1.ColumnHeaders. _
```



```

Add(, , "Section", Form2.ListView1.Width / 4)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "Minimum Sheave Diameter", Form2.ListView1.Width / 4)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "Standard", Form2.ListView1.Width / 4)

'--- Sets the view style of the Listbox
Form2.ListView1.BorderStyle = ccFixedSingle
'--- Selects the data from the particular table
str = "select * from TABLE17_2"
'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
    If Not IsNull(rs(1)) Then
        itmX.SubItems(1) = CStr(rs(1))
    End If
    If Not IsNull(rs(2)) Then
        itmX.SubItems(2) = CStr(rs(2))
    End If
    If Not IsNull(rs(3)) Then
        itmX.SubItems(3) = CStr(rs(3))
    End If

    rs.MoveNext
Loop
'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show

```

```
'--- Set the form Caption
Form2.Caption = "STANDARD DESIGNATIONS FOR VARIOUS V-BELTS"
```

```
End Sub
```

```
*****
```

This function is executed on clicking the Table 17-4 menu item and displays the contents of the table

```
*****
```

```
Private Sub mnuT17_4_Click()
```

```
'--- Variable declaration for sql query and for accessing table data
```

```
Dim str As String
```

```
Dim rs As New ADODB.Recordset
```

```
'--- Assign the column headers for the ListBox
```

```
Dim itmX As ListItem
```

```
Dim clmX As ColumnHeader
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Belt Section", Form2.ListView1.Width / 3)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Size Range (in)", Form2.ListView1.Width / 3)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Conversion Quantity (in)", Form2.ListView1.Width / 3)
```

```
'--- Sets the view style of the Listbox
```

```
Form2.ListView1.BorderStyle = ccFixedSingle
```

```
'--- Selects the data from the particular table
```

```
str = "select * from TABLE17_4"
```

```
'--- Open Database connection
```

```
rs.Open str, connect
```

```
'--- Assign each row of the table to each row of the listbox
```

```

Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
    If Not IsNull(rs(1)) Then
        itmX.SubItems(1) = CStr(rs(1))
    End If
    If Not IsNull(rs(2)) Then
        itmX.SubItems(2) = CStr(rs(2))
    End If
    rs.MoveNext
Loop
'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption
Form2.Caption = "LENGTH CONVERSION QUANTITIES FOR HEAVY-DUTY
CONVENTIONAL INCH-SERIES BELTS"

```

```
End Sub
```

```
*****
```

This function is executed on clicking the Table 17-7 menu item and displays the contents of the table

```
*****
```

```
Private Sub mnuT17_7_Click()
```

```
'--- Variable declaration for sql query and for accessing table data
```

```
Dim str As String
```

```
Dim rs As New ADODB.Recordset
```

```
'--- Assign the column headers for the ListBox
```

```

Dim itmX As ListItem
Dim clmX As ColumnHeader
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "Ls", Form2.ListView1.Width / 5)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "A", Form2.ListView1.Width / 5)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "B", Form2.ListView1.Width / 5)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "C", Form2.ListView1.Width / 5)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "D", Form2.ListView1.Width / 5)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "E", Form2.ListView1.Width / 5)

'--- Sets the view style of the Listbox
Form2.ListView1.BorderStyle = ccFixedSingle
'--- Selects the data from the particular table
str = "select * from TABLE17_7"
'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, CStr(rs(0)))
    If Not IsNull(rs(1)) Then
        itmX.SubItems(1) = CStr(rs(1))
    End If
    If Not IsNull(rs(2)) Then
        itmX.SubItems(2) = CStr(rs(2))
    End If
    If Not IsNull(rs(3)) Then

```

```

        itmX.SubItems(3) = CStr(rs(3))
    End If
    If Not IsNull(rs(4)) Then
        itmX.SubItems(4) = CStr(rs(4))
    End If
    If Not IsNull(rs(5)) Then
        itmX.SubItems(5) = CStr(rs(5))
    End If
    rs.MoveNext
Loop

'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption
Form2.Caption = "STANDARD LENGTHS Ls AND LENGTH-CORRECTION
FACTORS K2 FOR HEAVY-DUTY CONVENTIONAL ENGLISH V-BELTS"

End Sub

*****
This function is executed on clicking the Table 17-9 menu item and displays the contents
of the table
*****

Private Sub mnuT17_9_Click()

'--- Variable declaration for sql query and for accessing table data
Dim str As String
Dim rs As New ADODB.Recordset

```

```

'--- Assign the column headers for the ListBox
Dim itmX As ListItem
Dim clmX As ColumnHeader
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "Driven Machinery", Form2.ListView1.Width / 3)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "Normal Torque Characteristic", Form2.ListView1.Width / 3)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, "High or Non-Uniform Torque", Form2.ListView1.Width / 3)
'--- Sets the view style of the Listbox
Form2.ListView1.BorderStyle = ccFixedSingle
'--- Selects the data from the particular table
str = "select * from TABLE17_9"
'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
    If Not IsNull(rs(1)) Then
        itmX.SubItems(1) = CStr(rs(1))
    End If
    If Not IsNull(rs(2)) Then
        itmX.SubItems(2) = CStr(rs(2))
    End If
    rs.MoveNext
Loop
'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption

```

```
Form2.Caption = "SUGGESTED SERVICE FACTORS Ks FOR V-BELT DRIVES"
```

```
End Sub
```

```
*****
```

This function is executed on clicking the Table 17-3 menu item and displays the contents of the table

```
*****
```

```
Private Sub mnuT17_3_Click()
```

```
'--- Variable declaration for sql query and for accessing table data
```

```
Dim str As String
```

```
Dim rs As New ADODB.Recordset
```

```
'--- Assign the column headers for the ListBox
```

```
Dim itmX As ListItem
```

```
Dim clmX As ColumnHeader
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Belt section", Form2.ListView1.Width / 5)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Width, a (in)", Form2.ListView1.Width / 5)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Single Belt Thickness, b (in)", Form2.ListView1.Width / 5)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Joined multiple belt thickness, b'(in)", Form2.ListView1.Width / 5)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Power Range per belt (hp)", Form2.ListView1.Width / 5)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Typical Standard Sheave Sizes (in)", Form2.ListView1.Width / 5)
```

```
'--- Sets the view style of the Listbox
```

```

Form2.ListView1.BorderStyle = ccFixedSingle
'--- Selects the data from the particular table
str = "select * from TABLE17_3"
'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
    If Not IsNull(rs(1)) Then
        itmX.SubItems(1) = CStr(rs(1))
    End If
    If Not IsNull(rs(2)) Then
        itmX.SubItems(2) = CStr(rs(2))
    End If
    If Not IsNull(rs(3)) Then
        itmX.SubItems(3) = CStr(rs(3))
    End If
    If Not IsNull(rs(4)) Then
        itmX.SubItems(4) = CStr(rs(4))
    End If
    If Not IsNull(rs(5)) Then
        itmX.SubItems(5) = CStr(rs(5))
    End If
    rs.MoveNext
Loop
'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption
Form2.Caption = "HEAVY-DUTY CONVENTIONAL V-BELT SECTIONS"

```


End Sub

This function is executed on clicking the Table 17-5 menu item and displays the contents of the table

Private Sub mnuT17_5_Click()

'--- Variable declaration for sql query and for accessing table data

Dim str As String

Dim rs As New ADODB.Recordset

'--- Assign the column headers for the ListBox

Dim itmX As ListItem

Dim clmX As ColumnHeader

Set clmX = Form2.ListView1.ColumnHeaders. _

Add(, "Belt section", Form2.ListView1.Width / 5)

Set clmX = Form2.ListView1.ColumnHeaders. _

Add(, "C1", Form2.ListView1.Width / 5)

Set clmX = Form2.ListView1.ColumnHeaders. _

Add(, "C2", Form2.ListView1.Width / 5)

Set clmX = Form2.ListView1.ColumnHeaders. _

Add(, "C3", Form2.ListView1.Width / 5)

Set clmX = Form2.ListView1.ColumnHeaders. _

Add(, "C4", Form2.ListView1.Width / 5)

'--- Sets the view style of the Listbox

Form2.ListView1.BorderStyle = ccFixedSingle

'--- Selects the data from the particular table

str = "select * from TABLE17_5"

```

'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
  Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
    If Not IsNull(rs(1)) Then
      itmX.SubItems(1) = CStr(rs(1))
    End If
    If Not IsNull(rs(2)) Then
      itmX.SubItems(2) = CStr(rs(2))
    End If
    If Not IsNull(rs(3)) Then
      itmX.SubItems(3) = CStr(rs(3))
    End If
    If Not IsNull(rs(4)) Then
      itmX.SubItems(4) = CStr(rs(4))
    End If
    rs.MoveNext
  Loop
'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption
Form2.Caption = "CONSTANTS FOR USE IN THE POWER-RATING EQUATION"

End Sub

```

This function is executed on clicking the Table 17-6 menu item and displays the contents of the table

```
Private Sub mnuT17_6_Click()
```

```
'--- Variable declaration for sql query and for accessing table data
```

```
Dim str As String
```

```
Dim rs As New ADODB.Recordset
```

```
'--- Assign the column headers for the ListBox
```

```
Dim itmX As ListItem
```

```
Dim clmX As ColumnHeader
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "D/d Range", Form2.ListView1.Width / 2)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Ka", Form2.ListView1.Width / 2)
```

```
'--- Sets the view style of the Listbox
```

```
Form2.ListView1.BorderStyle = ccFixedSingle
```

```
'--- Selects the data from the particular table
```

```
str = "select * from TABLE17_6"
```

```
'--- Open Database connection
```

```
rs.Open str, connect
```

```
'--- Assign each row of the table to each row of the listbox
```

```
Do While Not rs.EOF
```

```
Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
```

```
If Not IsNull(rs(1)) Then
```

```
itmX.SubItems(1) = CStr(rs(1))
```

```
End If
```

```
rs.MoveNext
```

```
Loop
```

```

'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption
Form2.Caption = "SPEED-RATIO FACTORS FOR USE IN THE POWER-RATING
                EQUATION"

```

End Sub

```

*****

```

This function is executed on clicking the Table 17-10 menu item and displays the contents of the table

```

*****

```

Private Sub mnuTable17_10_Click()

```

'--- Variable declaration for sql query and for accessing table data

```

```

Dim str As String

```

```

Dim rs As New ADODB.Recordset

```

```

'--- Assign the column headers for the ListBox

```

```

Dim itmX As ListItem

```

```

Dim clmX As ColumnHeader

```

```

Set clmX = Form2.ListView1.ColumnHeaders. _

```

```

Add(, "ANSI Chain Number", Form2.ListView1.Width / 7)

```

```

Set clmX = Form2.ListView1.ColumnHeaders. _

```

```

Add(, "Pitch (in)", Form2.ListView1.Width / 7)

```

```

Set clmX = Form2.ListView1.ColumnHeaders. _

```

```

Add(, "Width(in)", Form2.ListView1.Width / 7)

```

```

Set clmX = Form2.ListView1.ColumnHeaders. _

```

```

Add(, , "Minimum Tensile Strength (lb)", Form2.ListView1.Width / 7)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "Average weight(lb/ft)", Form2.ListView1.Width / 7)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "Roller Diameter(in)", Form2.ListView1.Width / 7)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "Multiple Strand Spacing(in)", Form2.ListView1.Width / 7)
'--- Sets the view style of the Listbox
Form2.ListView1.BorderStyle = ccFixedSingle
'--- Selects the data from the particular table
str = "select * from TABLE17_10"
'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
Do While Not rs.EOF
    Set itmx = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
        If Not IsNull(rs(1)) Then
            itmx.SubItems(1) = CStr(rs(1))
        End If
        If Not IsNull(rs(2)) Then
            itmx.SubItems(2) = CStr(rs(2))
        End If
        If Not IsNull(rs(3)) Then
            itmx.SubItems(3) = CStr(rs(3))
        End If
        If Not IsNull(rs(4)) Then
            itmx.SubItems(4) = CStr(rs(4))
        End If
        If Not IsNull(rs(5)) Then
            itmx.SubItems(5) = CStr(rs(5))
        End If

```

```

        If Not IsNull(rs(6)) Then
            itmX.SubItems(6) = CStr(rs(6))
        End If
        rs.MoveNext
    Loop
    '--- Set the type of view to Reportview
    Form2.ListView1.View = lvwReport
    '--- Show the form
    Form2.Show
    '--- Set the form Caption
    Form2.Caption = "RATED HORSEPOWER CAPACITY OF SINGLE STRAND
                    SINGLE PITCH ROLLER CHAIN FOR 17 TOOTH SPROCKET"

End Sub

*****
This function is executed on clicking the Table 17-11 menu item and displays the
contents of the table
*****

Private Sub mnuTable17_11_Click()

    '--- Variable declaration for sql query and for accessing table data
    Dim str As String
    Dim rs As New ADODB.Recordset
        '--- Assign the column headers for the ListBox
        Dim itmX As ListItem
        Dim clmX As ColumnHeader

        Set clmX = Form2.ListView1.ColumnHeaders. _
            Add(, "Sprocket Speed(rpm)", Form2.ListView1.Width / 15)

```

```

Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "25", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "35", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "40", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "41", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "50", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "60", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "80", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "100", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "120", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "140", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "160", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "180", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "200", Form2.ListView1.Width / 15)
Set clmX = Form2.ListView1.ColumnHeaders. _
Add(, , "240", Form2.ListView1.Width / 15)

```

```
'--- Sets the view style of the Listbox
```

```
Form2.ListView1.BorderStyle = ccFixedSingle
```

```

'--- Selects the data from the particular table
str = "select * from TABLE17_11"
'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
        If Not IsNull(rs(1)) Then
            itmX.SubItems(1) = CStr(rs(1))
        End If
        If Not IsNull(rs(2)) Then
            itmX.SubItems(2) = CStr(rs(2))
        End If
        If Not IsNull(rs(3)) Then
            itmX.SubItems(3) = CStr(rs(3))
        End If
        If Not IsNull(rs(4)) Then
            itmX.SubItems(4) = CStr(rs(4))
        End If
        If Not IsNull(rs(5)) Then
            itmX.SubItems(5) = CStr(rs(5))
        End If
        If Not IsNull(rs(6)) Then
            itmX.SubItems(6) = CStr(rs(6))
        End If
        If Not IsNull(rs(7)) Then
            itmX.SubItems(7) = CStr(rs(7))
        End If
        If Not IsNull(rs(8)) Then
            itmX.SubItems(8) = CStr(rs(8))
        End If
    End While

```



```

    If Not IsNull(rs(9)) Then
        itmx.SubItems(9) = CStr(rs(9))
    End If
    If Not IsNull(rs(10)) Then
        itmx.SubItems(10) = CStr(rs(10))
    End If
    If Not IsNull(rs(11)) Then
        itmx.SubItems(11) = CStr(rs(11))
    End If
    If Not IsNull(rs(12)) Then
        itmx.SubItems(12) = CStr(rs(12))
    End If
    If Not IsNull(rs(13)) Then
        itmx.SubItems(13) = CStr(rs(13))
    End If
    If Not IsNull(rs(14)) Then
        itmx.SubItems(14) = CStr(rs(14))
    End If
    rs.MoveNext
Loop
'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption
Form2.Caption = "RATED HORSEPOWER CAPACITY OF SINGLE STRAND
                SINGLE PITCH ROLLER CHAIN FOR 17 TOOTH SPROCKET"

End Sub

```

This function is executed on clicking the Table 17-12 menu item and displays the contents of the table

```
Private Sub mnuTable17_12_Click()
```

```
'--- Variable declaration for sql query and for accessing table data
```

```
Dim str As String
```

```
Dim rs As New ADODB.Recordset
```

```
'--- Assign the column headers for the ListBox
```

```
Dim itmX As ListItem
```

```
Dim clmX As ColumnHeader
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Number of teeth on driving sprocket", Form2.ListView1.Width / 2)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Tooth Correction factor K1", Form2.ListView1.Width / 2)
```

```
'--- Sets the view style of the Listbox
```

```
Form2.ListView1.BorderStyle = ccFixedSingle
```

```
'--- Selects the data from the particular table
```

```
str = "select * from TABLE17_12"
```

```
'--- Open Database connection
```

```
rs.Open str, connect
```

```
'--- Assign each row of the table to each row of the listbox
```

```
Do While Not rs.EOF
```

```
Set itmX = Form2.ListView1.ListItems.Add(, CStr(rs(0)))
```

```
If Not IsNull(rs(1)) Then
```

```
itmX.SubItems(1) = CStr(rs(1))
```

```
End If
```

```
rs.MoveNext
```

```

    Loop
    '--- Set the type of view to Reportview
    Form2.ListView1.View = lvwReport
    '--- Show the form
    Form2.Show
    '--- Set the form Caption
    Form2.Caption = "TOOTH CORRECTION FACTORS"

End Sub

*****
This function is executed on clicking the Table 17-13 menu item and displays the
contents of the table
*****

Private Sub mnuTable17_13_Click()

'--- Variable declaration for sql query and for accessing table data
Dim str As String
Dim rs As New ADODB.Recordset
    '--- Assign the column headers for the ListBox
    Dim itmX As ListItem
    Dim clmX As ColumnHeader
    Set clmX = Form2.ListView1.ColumnHeaders. _
    Add(, "Number of strands", Form2.ListView1.Width / 2)
    Set clmX = Form2.ListView1.ColumnHeaders. _
    Add(, "K2", Form2.ListView1.Width / 2)

    '--- Sets the view style of the Listbox
    Form2.ListView1.BorderStyle = ccFixedSingle
    '--- Selects the data from the particular table

```

```

str = "select * from TABLE17_13"
'--- Open Database connection
rs.Open str, connect
'--- Assign each row of the table to each row of the listbox
Do While Not rs.EOF
    Set itmX = Form2.ListView1.ListItems.Add(, , CStr(rs(0)))
    If Not IsNull(rs(1)) Then
        itmX.SubItems(1) = CStr(rs(1))
    End If
    rs.MoveNext
Loop
'--- Set the type of view to Reportview
Form2.ListView1.View = lvwReport
'--- Show the form
Form2.Show
'--- Set the form Caption
Form2.Caption = "MULTIPLE STRAND FACTORS K2"

End Sub

*****
This function is executed on clicking the Table 17-14 menu item and displays the
contents of the table
*****

Private Sub mnuTable17_14_Click()

'--- Variable declaration for sql query and for accessing table data
Dim str As String
Dim rs As New ADODB.Recordset

'--- Assign the column headers for the ListBox

```

```
Dim itmX As ListItem
```

```
Dim clmX As ColumnHeader
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Driven Machinery", Form2.ListView1.Width / 4)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Internal Combustion Engine with hydraulic drive", Form2.ListView1.Width /
```

```
4)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Electric Motor or turbine", Form2.ListView1.Width / 4)
```

```
Set clmX = Form2.ListView1.ColumnHeaders. _
```

```
Add(, "Internal combustion engine with mechanical drive", Form2.ListView1.Width /
```

```
4)
```

```
'--- Sets the view style of the Listbox
```

```
Form2.ListView1.BorderStyle = ccFixedSingle
```

```
'--- Selects the data from the particular table
```

```
str = "select * from TABLE17_14"
```

```
'--- Open Database connection
```

```
rs.Open str, connect
```

```
'--- Assign each row of the table to each row of the listbox
```

```
Do While Not rs.EOF
```

```
    Set itmX = Form2.ListView1.ListItems.Add(, CStr(rs(0)))
```

```
        If Not IsNull(rs(1)) Then
```

```
            itmX.SubItems(1) = CStr(rs(1))
```

```
        End If
```

```
        If Not IsNull(rs(2)) Then
```

```
            itmX.SubItems(2) = CStr(rs(2))
```

```
        End If
```

```
        If Not IsNull(rs(3)) Then
```

```
            itmX.SubItems(3) = CStr(rs(3))
```

```

        End If
        rs.MoveNext
    Loop
    '--- Set the type of view to Reportview
    Form2.ListView1.View = lvwReport
    '--- Show the form
    Form2.Show
    '--- Set the form Caption
    Form2.Caption = "LOAD SERVICE FACTORS Ks"

End Sub

```

```

*****
frmTable.frm
This is the source code for Close button on the form displaying the tables
*****

```

```

Private Sub cmdClose_Click()

    '--- Closes the form
    Unload Me

End Sub

```

```

*****
frmSpecs.frm
This is the source code for the form displaying the specifications.
The specifications input by the designer are assigned to the respective labels in the main
form
*****

```

```
Private Sub cmdOK_Click()
```

```
    For i = 1 To 6
```

```
        Form1.lblSpec(i).Caption = Form3.TxtSpec(i).Text
```

```
    Next i
```

```
    '--- Show Form1
```

```
    Form1.Show
```

```
    '--- Closes the form
```

```
    Unload Me
```

```
End Sub
```

```
*****
```

```
frmGraphs.frm
```

```
This is the source code for the form displaying graphs.
```

```
*****
```

```
Private Sub cmdClose_Click()
```

```
    '--- Closes the form
```

```
    Unload Me
```

```
End Sub
```

```
*****
```

```
frmBrowser.frm
```

```
This is the source code for the form displaying the browser.
```

```
*****
```

```
'--- Variable to hold starting address
```

```
Public StartingAddress As String
```

```
'--- Variable to check for navigation
Dim mbDontNavigateNow As Boolean
```

```
*****
```

```
frmBrowser.frm
```

```
This is the source code for the form load event
```

```
*****
```

```
Private Sub Form_Load()
```

```
    On Error Resume Next
```

```
    '--- Show the form
```

```
    Me.Show
```

```
    '--- Refresh the toolbar
```

```
    tbToolBar.Refresh
```

```
    '--- Resize the form
```

```
    Form_Resize
```

```
    '--- Specify the position of the address
```

```
    cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15
```

```
    If Len(StartingAddress) > 0 Then
```

```
        cboAddress.Text = StartingAddress
```

```
        cboAddress.AddItem cboAddress.Text
```

```
        '--- Try to navigate to the starting address
```

```
        timTimer.Enabled = True
```

```
        brwWebBrowser.Navigate StartingAddress
```

```
    End If
```

```
End Sub
```

This function is executed on completion of the download of a webpage

```
Private Sub brwWebBrowser_DownloadComplete()
```

```
    On Error Resume Next
```

```
    '--- Assign the location name to the form caption
```

```
    Me.Caption = brwWebBrowser.LocationName
```

```
End Sub
```

This function checks the web address in the combo box. If an address is present, it navigates to the given address

```
Private Sub brwWebBrowser_NavigateComplete2(ByVal pDisp As Object, URL As Variant)
```

```
    On Error Resume Next
```

```
    Dim i As Integer
```

```
    Dim bFound As Boolean
```

```
    Me.Caption = brwWebBrowser.LocationName
```

```
    '--- If an address exists within the address combo box, then navigate to the address
```

```
    For i = 0 To cboAddress.ListCount - 1
```

```
        If cboAddress.List(i) = brwWebBrowser.LocationURL Then
```

```
            bFound = True
```

```
            Exit For
```

```
        End If
```

```
    Next i
```

```
mbDontNavigateNow = True
If bFound Then
    cboAddress.RemoveItem i
End If
cboAddress.AddItem brwWebBrowser.LocationURL, 0
cboAddress.ListIndex = 0
mbDontNavigateNow = False
```

End Sub

This function navigates to the address clicked in the combo box

Private Sub cboAddress_Click()

```
    If mbDontNavigateNow Then Exit Sub
    '--- Enable timer
    timTimer.Enabled = True
    '--- Navigate to address
    brwWebBrowser.Navigate cboAddress.Text
```

End Sub

This function responds to the key press event (enter key) within the combo box.

Private Sub cboAddress_KeyPress(KeyAscii As Integer)

```
    On Error Resume Next
    '--- If enter key is pressed
```

```
If KeyAscii = vbKeyReturn Then
    '--- Call cboAddress_Click
    cboAddress_Click
End If
```

```
End Sub
```

```
*****
```

```
This function resizes the browser form.
```

```
*****
```

```
Private Sub Form_Resize()
```

```
    On Error Resume Next
```

```
    '--- Adjust the address combo box width
```

```
    cboAddress.Width = Me.ScaleWidth - 100
```

```
    '--- Adjust the width of the form
```

```
    brwWebBrowser.Width = Me.ScaleWidth - 100
```

```
    '--- Adjust the height of the form
```

```
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height) -
100
```

```
End Sub
```

```
*****
```

```
This function is for the timer control on the browser form
```

```
*****
```

```
Private Sub timTimer_Timer()
```

```

If brwWebBrowser.Busy = False Then
    '--- Enable timer
    timTimer.Enabled = False
    '--- Assign loaded location name to the form caption
    Me.Caption = brwWebBrowser.LocationName
Else
    '--- Assign working to the form caption
    Me.Caption = "Working..."
End If

```

```
End Sub
```

```
*****
```

This function is executed on clicking any of the tool bar buttons

```
*****
```

```
Private Sub tbToolBar_ButtonClick(ByVal Button As Button)
```

```

    On Error Resume Next
    '--- Enable the timer
    timTimer.Enabled = True
    '--- Based on the button selected perform the navigations
    Select Case Button.Key
        Case "Back"
            brwWebBrowser.GoBack
        Case "Forward"
            brwWebBrowser.GoForward
        Case "Refresh"
            brwWebBrowser.Refresh
        Case "Home"
            brwWebBrowser.GoHome
    End Select

```

```

Case "Search"
    brwWebBrowser.GoSearch
Case "Stop"
    timTimer.Enabled = False
    brwWebBrowser.Stop
    Me.Caption = brwWebBrowser.LocationName
End Select

```

```
End Sub
```

```
*****
```

```
frmFile.frm
```

```
This is the source code for the form displaying the contents of a file on the desktop
```

```
*****
```

```
Private Sub Form_Resize()
```

```

    Dim intMsg As Integer '---For MsgBox()
    '--- Change the size of the list box if the form is resized
    '--- Make sure the form is not sized so small that the list box cannot display
    If (frmFile.Width < 400) Or (frmFile.Height < 3500) Then
        '--- Hide list box and warn
        lstFile.Visible = False
        intMsg = MsgBox("The form is too small to display the file", vbCritical)
    Else
        '--- Turn on list box display in case it was turned off previously
        lstFile.Visible = True
        '--- Adjust the size of the list box to the form
        lstFile.Width = frmFile.Width - 1440
        lstFile.Height = frmFile.Height - 2500
    End If

```

End Sub

This function is executed on clicking the Exit menu item

Private Sub mnuFileExit_Click()

'--- Unloads the form

Unload Me

End Sub

This function is executed on clicking the Open menu item and displays the contents of the file

Private Sub mnuFileOpen_Click()

Dim strFileLine As String

'--- Set up for cancel click

On Error GoTo comErrorHandler

'--- Display the file open dialogue box

comFile.ShowOpen

'--- Code either continues if the clicked or skips to error handler if cancel clicked

'--- Open the file the selected

Open comFile.FileName For Input As #1

'--- Make room for new file

lstFile.Clear

```

'--- Read one complete line of file
Line Input #1, strFileLine
lstFile.AddItem strFileLine
'--- Keep reading and adding to the list box until end of file is reached
Do Until (EOF(1))
    Line Input #1, strFileLine
    lstFile.AddItem strFileLine
Loop
'--- Close open file
Close
comErrorHandler:
    '--- Do nothing if the cancel is clicked

End Sub

*****

frmInputInfo
This is the source code for the form to input information into a file on the desktop.
*****

Private Sub cmdFolders_Click()

'--- Variable for storing the file number
Dim iFileNum As Integer
'--- Variable to hold the file path name
Dim filePath As String
'--- Variable to hold the directory path name
Dim dirPath As String
'--- Variable to hold the text
Dim textTowrite As String

```

```

'---Assigns the text box text to folder name
dirPath = txtDir.Text
'---Assigns the textbox text to file name
filePath = txtFile.Text
'---Assigns the textbox text to the file
textTowrite = txtWrite.Text
'---Check if directory already exists
If CreateNewDirectory(dirPath) = True Then
    iFileNum = FreeFile
    '--- Gets next available FileNumber in case other files are open
    '--- Open "File Name" for Output as #iFileNum
    filePath = dirPath & "\" & filePath
    Open filePath For Output As #iFileNum
    Print #iFileNum, textTowrite
    Close #iFileNum
Else
    '--- An Error has occurred
End If
'--- Show Form1
Form1.Show

End Sub

```

The following function creates the folder and file dynamically on the desktop

```

Private Function CreateNewDirectory(sPath As String) As Boolean
    '--- Default return value is false
    CreateNewDirectory = False
    '--- Use inline error trapping

```



```

On Error Resume Next
'--- Change to requested drive
ChDrive sPath
'--- Check for errors
If Err Then
    '--- No access to assigned drive
    MsgBox Error$, vbExclamation
    Exit Function
End If
'--- Change to requested directory
ChDir sPath
'--- Check for errors
If Err = 0 Then
    '--- Directory already exists
    CreateNewDirectory = True
Else
    '--- Path does not currently exist so create it
    Err.Clear
    '--- Create new directory
    MkDir sPath
    '--- Check for errors
    If Err = 0 Then
        '--- Success
        CreateNewDirectory = True
    Else
        '--- Cannot create new directory
        MsgBox "Cannot create new directory." & vbCrLf & vbCrLf & Err & vbCrLf &
            Error$, vbExclamation
    End If
End If

```

End Function

Module

This is the code module containing all the global declarations

Option Explicit

'--- For establishing active database connectivity

Public connect As ADODB.Connection

'--- For establishing the record set

Public rs As ADODB.Recordset

'--- Counter for the generation of textboxes

Public count1 As Integer

'--- For the dynamic generation of textboxes

Public gendata() As String

'--- Establishing the object for querying the database

Public sql As String

'--- For establishing the OLE link to TK Solver

Public tkw As Object

'--- Variable for driven sheave diameter, D

Global BeltNewD1 As Double

'--- Variable for driving sheave diameter, d

Global BeltNewD2 As Double

'--- Variable for center distance, C

Global BeltCen As Double

'--- Variable for power rating, Hr

Global BeltHr As Double

'--- Variable for corrected power rating, H_r

Global BeltH_r As Double

'--- Variable for design power, Pd

Global BeltPd As Double

'--- Variable for belt length, Lp

Global BeltLp As Double