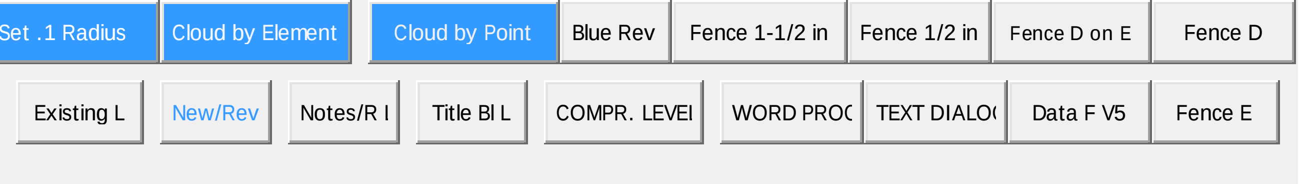
```
Private Sub BackCircle_Click()
 ' MsgBox "hello world"
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Backcircle"""

    CadInputQueue.SendCommand "ACTIVE WEIGHT 3"
'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 1, "MGDSHOOK"

'   Start a command
    CadInputQueue.SendCommand "PLACE REVCLOUD ELEMENT"
End Sub

Private Sub BCRadius_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long
    Dim TxtHeight As Long

    'ActiveSettings.TextStyle.Height
    TxtHeight = ActiveSettings.TextStyle.Height

'   Start a command
    CadInputQueue.SendCommand "PLACE REVCLOUD POINTS"

'   Set a variable associated with a dialog box
    SetCExpressionValue "cloudParams.flags.lockRadius", 1, "COMPCURV"

    'SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * 0.1), "COMPCURV"

    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * ActiveSettings.TextStyle.Height), "COMPCURV"

    CommandState.StartDefaultCommand
End Sub



Private Sub Existing_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Existing"""

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 0, "MGDSHOOK"

    CadInputQueue.SendCommand "ACTIVE STYLE 0"
    CadInputQueue.SendCommand "ACTIVE WEIGHT 1"

    CommandState.StartDefaultCommand
End Sub

Private Sub NewRev_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""New or Revisions"""

    CadInputQueue.SendCommand "ACTIVE STYLE 0"

    CadInputQueue.SendCommand "ACTIVE WEIGHT 1"

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 7, "MGDSHOOK"

    CommandState.StartDefaultCommand
End Sub

Private Sub Spare_Click()

End Sub

Private Sub Notes_Click()
 Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Notes and References"""

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 7, "MGDSHOOK"

    CommandState.StartDefaultCommand
End Sub
```

Existing | New / Rev | BackCircle | BC Radius | | Notes | F2 Checks

```
 '**********Get Resolution and Set General Text Height***********
  ' Dim NoErrorVal As Boolean
  ' NoErrorVal = True
  ' msg = GetResolSetGenTextHgt(NoErrorVal)
  ' If NoErrorVal = True Then
     'set cloud radius
     'following code does not work for some reason
     ' SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * ActiveSettings.TextStyle.Height), "COMPCU
RV"
  ' End If
  ' rMsg = rMsg & msg


Private Sub cmbCloudEl_Click()
 ' MsgBox "hello world"
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Backcircle"""

    CadInputQueue.SendCommand "ACTIVE WEIGHT 3"
'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 1, "MGDSHOOK"

'   Start a command
    CadInputQueue.SendCommand "PLACE REVCLOUD ELEMENT"

'    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * 0.1), "COMPCURV"

'    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * ActiveSettings.TextStyle.Height), "COMPCURV
"


End Sub

Private Sub cmbCloudPt_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Backcircle"""

    CadInputQueue.SendCommand "ACTIVE WEIGHT 3"

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 1, "MGDSHOOK"

'    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * 0.1), "COMPCURV"


'   Start a command
```

```
    CadInputQueue.SendCommand "PLACE REVCLOUD POINTS"

End Sub




Private Sub cmbExistingLevel_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Existing"""

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 0, "MGDSHOOK"

    CadInputQueue.SendCommand "ACTIVE STYLE 0"

    CadInputQueue.SendCommand "ACTIVE WEIGHT 1"

    CommandState.StartDefaultCommand
End Sub




Private Sub cmdEsizeFence_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"

'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.493826490298015, 30.0246028833885, 0#), 1

'   Coordinates are in master units
    startPoint.X = 0#
    startPoint.Y = 30#
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(41.97131768619, 1.17293206833481E-02, 0#), 1
    point.X = startPoint.X + 42.0000000000001
    point.Y = startPoint.Y - 30#
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X + 1.79290742839691
    point.Y = startPoint.Y + 3.46071529917275
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "PRINT MAXIMIZE"

    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 40.9646372074583
    point.Y = startPoint.Y - 30.3439460111004
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 40.801902248909
    point.Y = startPoint.Y - 20.3922465283965
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub

Private Sub cmdFence18000_Click()
  Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"

'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.098639241090924, 14.4992497162524, 1.4111111111138), 1

'   Coordinates are in master units
    startPoint.X = 0#
    startPoint.Y = 14.6666666666666
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(22.6630469909838, -4.14684863753751E-03, 1.41111111111402), 1

    point.X = startPoint.X + 22.6666666666667
    point.Y = startPoint.Y - 14.6666666666666
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X + 23.1448617967697
    point.Y = startPoint.Y - 6.5622857142857
    point.Z = startPoint.Z + 1.4111111111123
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "FIT VIEW EXTENDED 5"

    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 21.7245776434224
    point.Y = startPoint.Y - 8.34410349586891
    point.Z = startPoint.Z + 1.41111111111421
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 21.6370483654411
    point.Y = startPoint.Y - 14.7128887948723
    point.Z = startPoint.Z + 1.41111111111426
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand

End Sub




Private Sub cmdNewOrRevLevel_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""New or Revisions"""

    CadInputQueue.SendCommand "ACTIVE STYLE 0"

    CadInputQueue.SendCommand "ACTIVE WEIGHT 1"

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 7, "MGDSHOOK"

    CommandState.StartDefaultCommand
End Sub

Private Sub cmdSetRadius_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long
    Dim TxtHeight As Long

    'ActiveSettings.TextStyle.Height
    TxtHeight = ActiveSettings.TextStyle.Height

'   Start a command
    CadInputQueue.SendCommand "PLACE REVCLOUD POINTS"

'   Set a variable associated with a dialog box
    SetCExpressionValue "cloudParams.flags.lockRadius", 1, "COMPCURV"

    'SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * 0.1), "COMPCURV"

    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * ActiveSettings.TextStyle.Height), "COMPCURV"

    CommandState.StartDefaultCommand

End Sub


Private Sub cmdTextDialog_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro1ModalHandler5
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Preferences [descartes]"

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD USERPREF"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD SPELLCHECK"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub

Private Sub cmdTitleBlock_Click()
  Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command

    CadInputQueue.SendCommand "ACTIVE LEVEL ""Border-titleblock"""

'   Set a variable associated with a dialog box
   ' SetCExpressionValue "tcb->symbology.color", -1, "MGDSHOOK"

    'CadInputQueue.SendCommand "ACTIVE WEIGHT 0"

    CommandState.StartDefaultCommand
End Sub

Private Sub cmdWordProcessor_Click()
```

```
  Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro2ModalHandler1
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Preferences [descartes]"

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD USERPREF"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD SPELLCHECK"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub


Private Sub CommandButton1_Click()
 Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Notes and References"""

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 7, "MGDSHOOK"

    CommandState.StartDefaultCommand
End Sub



Private Sub UserForm_Click()

End Sub
```

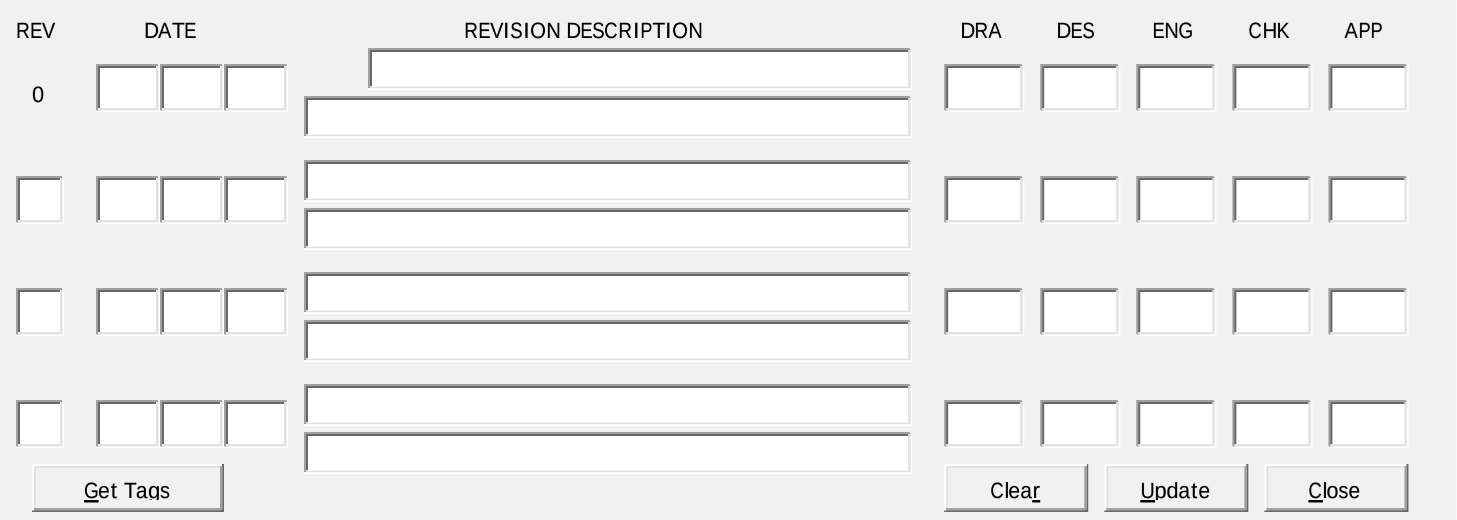

Set .1 Radius
Cloud by Element
Cloud by Point
Blue Rev
Fence 1-1/2 in
Fence 1/2 in
Fence D on E
Fence D
Existing L
New/Rev
Notes/R I
Title Bl L
COMPR. LEVEI
WORD PROC
TEXT DIALO(
Data F V5
Fence E

```
Private Sub cmdFolderPath_Click()

    'Get the folder of drawings and insert the path into the form
    lblFolderPath.Caption = modNextFile.SelectDGNFolder

    If lblFolderPath.Caption = "" Then
      MsgBox "You pressed Cancel, or you did not select a file inside of the folder."
      Exit Sub
    End If

    'save path to desktop\Filelists\path.txt
    modNextFile.WritePathToFile FullFolderPath:=lblFolderPath.Caption

    'create a text file of all of the DGN files in the folder
    Call modNextFile.FileCreate(lblFolderPath)

    'Create an array from the text file and insert the drawing names into
    'the listbox
    Call modNextFile.InsertFileLinesToArray(modNextFile.GetFileListPath)

End Sub


Private Sub ListBox1_Click()

    Dim path As String
    Dim filename As String
    Dim ReadOnly As Boolean

    path = lblFolderPath.Caption
    'filename = ListBox1.Selected(pvargindex)
    filename = ListBox1.Value
    'MsgBox filename
    filename = path & filename

    If togReadOnly.Caption = "Read Only" Then
      ReadOnly = True
     Else
      ReadOnly = False
    End If

    modNextFile.OpenFile filename:=filename, EditMode:=ReadOnly

    If ReadOnly = True Then
      'extent view 1
      CadInputQueue.SendCommand "FIT VIEW EXTENDED 1"
    End If

     If ReadOnly Then
       'MsgBox "This is read only mode"
     End If

End Sub
```

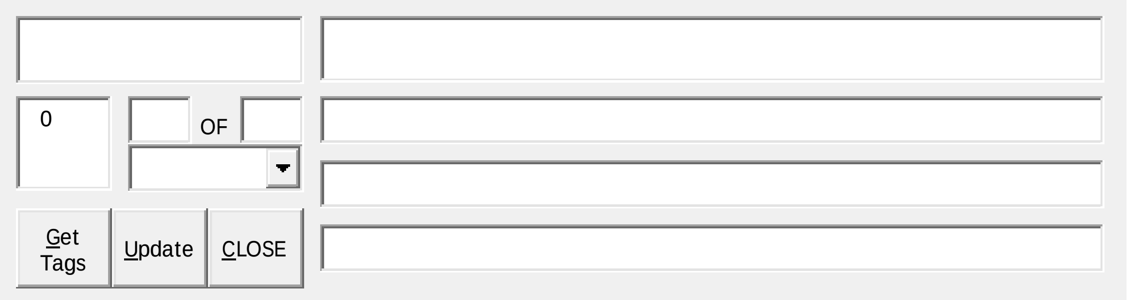
```
Private Sub ListBox1_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

   Select Case KeyAscii
     Case vbKeyF7
       MsgBox "f7 pressed"
     Case Else
       'do nothing
       MsgBox "some other key was pressed, not f7"
   End Select

   MsgBox "what"

End Sub

Private Sub togReadOnly_Click()
    If togReadOnly.Caption = "Read Only" Then
      togReadOnly.Caption = "Edit Mode"
     Else
      togReadOnly.Caption = "Read Only"
   End If
End Sub



Private Sub UserForm_Initialize()
  togReadOnly.Caption = "Read Only"
End Sub

Sub IncrementDown()

   Dim intCurSelect As Integer

   ListBox1.SetFocus
   If ListBox1.ListCount = 0 Then
     MsgBox "List Box is Empty!" & vbCrLf & "To fill box, Click -> LIST current DIR."
     Exit Sub
   End If

   If ListBox1.ListIndex = -1 Then
        'select the first in the list
        intCurSelect = 0
        'Label1.Caption = "nothing selected"

      Else
        intCurSelect = ListBox1.ListIndex
        If intCurSelect = ListBox1.ListCount - 1 Then
             Label1.Caption = 1
           Else
             Label1.Caption = intCurSelect + 2
         End If

        'increment index and text to see if this is the
        'last item on the list, if so then set next to be the first item in list
        intCurSelect = intCurSelect + 1
        TooHigh = ListBox1.ListCount
        If intCurSelect = TooHigh Then
         intCurSelect = 0
        End If
    End If

        ListBox1.ListIndex = intCurSelect

        'open file
        Label1.Caption = ListBox1.Value

   End Sub

Private Sub UserForm_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    Select Case KeyAscii
    Case vbKeyF7
      MsgBox "f7 pressed"
    Case Else
      'do nothing
      MsgBox "some other key was pressed, not f7"
   End Select
   MsgBox "what"
End Sub

Private Sub UserForm_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
  Select Case KeyAscii
    Case vbKeyF7
      MsgBox "f7 pressed"
    Case Else
      'do nothing
      MsgBox "some other key was pressed, not f7"
   End Select
   MsgBox "what"
End Sub

Private Sub UserForm_KeyUp(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
  Select Case KeyAscii
    Case vbKeyF7
      MsgBox "f7 pressed"
    Case Else
      'do nothing
      MsgBox "some other key was pressed, not f7"
   End Select
   MsgBox "what"
End Sub
```

LIST current DIR.
Read Only

```
Private Sub btnClear_Click()
  MsgBox "UNDER CONSTRUCTION - THE PLAN IS TO CLEAR AN ENTIRE LINE IF THE REVISION NUMBER IS LEFT BLANK."
End Sub

Private Sub btnClose_Click()
  Unload Me
End Sub

Private Sub btnGetRevTags_Click()
   Dim GetRevInfo As RevInfo

   GetRevInfo = modRevBlock.GetRevInfo

   With frmRevBlock
    .Rev1 = GetRevInfo.Rev1
    .Rev2 = GetRevInfo.Rev2
    .Rev3 = GetRevInfo.Rev3
    .DateMonth = GetRevInfo.Date_MN
    .DateDay = GetRevInfo.Date_DY
    .DateYear = GetRevInfo.Date_YR
    .DateAMonth = GetRevInfo.DateA_MN
    .DateADay = GetRevInfo.DateA_DY
    .DateAYear = GetRevInfo.DateA_YR
    .DateBMonth = GetRevInfo.DateB_MN
    .DateBDay = GetRevInfo.DateB_DY
    .DateBYear = GetRevInfo.DateB_YR
    .DateCMonth = GetRevInfo.DateC_MN
    .DateCDay = GetRevInfo.DateC_DY
    .DateCYear = GetRevInfo.DateC_YR
    .Line1a = GetRevInfo.Line1a
    .line1b = GetRevInfo.line1b
    .Line2a = GetRevInfo.Line2a
    .Line2b = GetRevInfo.Line2b
    .Line3a = GetRevInfo.Line3a
    .Line3b = GetRevInfo.Line3b
    .Line4a = GetRevInfo.Line4a
    .Line4b = GetRevInfo.Line4b
    .IREG_DRA = GetRevInfo.Reg_DRA
    .IREG_DES = GetRevInfo.Reg_DES
    .IREG_ENG = GetRevInfo.Reg_ENG
    .IREG_CHK = GetRevInfo.Reg_CHK
    .IREG_APP = GetRevInfo.Reg_APP
    .IA_DRA = GetRevInfo.A_DRA
    .IA_DES = GetRevInfo.A_DES
    .IA_ENG = GetRevInfo.A_ENG
    .IA_CHK = GetRevInfo.A_CHK
    .IA_APP = GetRevInfo.A_APP
    .IB_DRA = GetRevInfo.B_DRA
    .IB_DES = GetRevInfo.B_DES
    .IB_ENG = GetRevInfo.B_ENG
    .IB_CHK = GetRevInfo.B_CHK
    .IB_APP = GetRevInfo.B_APP
    .IC_DRA = GetRevInfo.C_DRA
    .IC_DES = GetRevInfo.C_DES
    .IC_ENG = GetRevInfo.C_ENG
    .IC_CHK = GetRevInfo.C_CHK
    .IC_APP = GetRevInfo.C_APP
   End With

End Sub

Private Sub btnUpdate_Click()
  Dim X As Long

   Dim AllRevInfo(43, 2) As String

   AllRevInfo(0, 0) = "RevNo"
   AllRevInfo(0, 1) = "Rev1"
   AllRevInfo(0, 2) = frmRevBlock.Rev1

   AllRevInfo(1, 0) = "RevNo"
   AllRevInfo(1, 1) = "Rev2"
   AllRevInfo(1, 2) = frmRevBlock.Rev2

   AllRevInfo(2, 0) = "RevNo"
   AllRevInfo(2, 1) = "Rev3"
   AllRevInfo(2, 2) = frmRevBlock.Rev3

   AllRevInfo(3, 0) = "Date"
   AllRevInfo(3, 1) = "1_Month"
   AllRevInfo(3, 2) = frmRevBlock.DateMonth

   AllRevInfo(4, 0) = "Date"
   AllRevInfo(4, 1) = "2_Day"
   AllRevInfo(4, 2) = frmRevBlock.DateDay

   AllRevInfo(5, 0) = "Date"
   AllRevInfo(5, 1) = "3_Year"
   AllRevInfo(5, 2) = frmRevBlock.DateYear

   AllRevInfo(6, 0) = "DateA"
   AllRevInfo(6, 1) = "1_Month"
   AllRevInfo(6, 2) = frmRevBlock.DateAMonth

   AllRevInfo(7, 0) = "DateA"
   AllRevInfo(7, 1) = "2_Day"
   AllRevInfo(7, 2) = frmRevBlock.DateADay

   AllRevInfo(8, 0) = "DateA"
   AllRevInfo(8, 1) = "3_Year"
   AllRevInfo(8, 2) = frmRevBlock.DateAYear

   AllRevInfo(9, 0) = "DateB"
   AllRevInfo(9, 1) = "1_Month"
   AllRevInfo(9, 2) = frmRevBlock.DateBMonth

   AllRevInfo(10, 0) = "DateB"
   AllRevInfo(10, 1) = "2_Day"
   AllRevInfo(10, 2) = frmRevBlock.DateBDay

   AllRevInfo(11, 0) = "DateB"
   AllRevInfo(11, 1) = "3_Year"
   AllRevInfo(11, 2) = frmRevBlock.DateBYear

   AllRevInfo(12, 0) = "DateC"
   AllRevInfo(12, 1) = "1_Month"
   AllRevInfo(12, 2) = frmRevBlock.DateCMonth

   AllRevInfo(13, 0) = "DateC"
   AllRevInfo(13, 1) = "2_Day"
   AllRevInfo(13, 2) = frmRevBlock.DateCDay

   AllRevInfo(14, 0) = "DateC"
   AllRevInfo(14, 1) = "3_Year"
   AllRevInfo(14, 2) = frmRevBlock.DateCYear

   AllRevInfo(15, 0) = "RevisionDescriptions"
   AllRevInfo(15, 1) = "Line1a"
   AllRevInfo(15, 2) = frmRevBlock.Line1a

   AllRevInfo(16, 0) = "RevisionDescriptions"
   AllRevInfo(16, 1) = "Line1b"
   AllRevInfo(16, 2) = frmRevBlock.line1b

   AllRevInfo(17, 0) = "RevisionDescriptions"
   AllRevInfo(17, 1) = "Line2a"
   AllRevInfo(17, 2) = frmRevBlock.Line2a

   AllRevInfo(18, 0) = "RevisionDescriptions"
   AllRevInfo(18, 1) = "Line2b"
   AllRevInfo(18, 2) = frmRevBlock.Line2b

   AllRevInfo(19, 0) = "RevisionDescriptions"
   AllRevInfo(19, 1) = "Line3a"
   AllRevInfo(19, 2) = frmRevBlock.Line3a

   AllRevInfo(20, 0) = "RevisionDescriptions"
   AllRevInfo(20, 1) = "Line3b"
   AllRevInfo(20, 2) = frmRevBlock.Line3b

   AllRevInfo(21, 0) = "RevisionDescriptions"
   AllRevInfo(21, 1) = "Line4a"
   AllRevInfo(21, 2) = frmRevBlock.Line4a

   AllRevInfo(22, 0) = "RevisionDescriptions"
   AllRevInfo(22, 1) = "Line4b"
   AllRevInfo(22, 2) = frmRevBlock.Line4b

   AllRevInfo(23, 0) = "Initals reg"
   AllRevInfo(23, 1) = "1_Drawn"
   AllRevInfo(23, 2) = frmRevBlock.IREG_DRA
   AllRevInfo(24, 0) = "Initals reg"
   AllRevInfo(24, 1) = "2_Designer"
   AllRevInfo(24, 2) = frmRevBlock.IREG_DES

   AllRevInfo(25, 0) = "Initals reg"
   AllRevInfo(25, 1) = "3_Engineer"
   AllRevInfo(25, 2) = frmRevBlock.IREG_ENG

   AllRevInfo(26, 0) = "Initals reg"
   AllRevInfo(26, 1) = "4_Check"
   AllRevInfo(26, 2) = frmRevBlock.IREG_CHK

   AllRevInfo(27, 0) = "Initals reg"
   AllRevInfo(27, 1) = "5_Approved"
   AllRevInfo(27, 2) = frmRevBlock.IREG_APP

   AllRevInfo(28, 0) = "InitialsA_1"
   AllRevInfo(28, 1) = "1_Drawn"
   AllRevInfo(28, 2) = frmRevBlock.IA_DRA

   AllRevInfo(29, 0) = "InitialsA_1"
   AllRevInfo(29, 1) = "2_Designer"
   AllRevInfo(29, 2) = frmRevBlock.IA_DES

   AllRevInfo(30, 0) = "InitialsA_1"
   AllRevInfo(30, 1) = "3_Engineer"
   AllRevInfo(30, 2) = frmRevBlock.IA_ENG

   AllRevInfo(31, 0) = "InitialsA_1"
   AllRevInfo(31, 1) = "4_Check"
   AllRevInfo(31, 2) = frmRevBlock.IA_CHK

   AllRevInfo(32, 0) = "InitialsA_1"
   AllRevInfo(32, 1) = "5_Approved"
   AllRevInfo(32, 2) = frmRevBlock.IA_APP

   AllRevInfo(33, 0) = "InitialsB_1"
   AllRevInfo(33, 1) = "1_Drawn"
   AllRevInfo(33, 2) = frmRevBlock.IB_DRA

   AllRevInfo(34, 0) = "InitialsB_1"
   AllRevInfo(34, 1) = "2_Designer"
   AllRevInfo(34, 2) = frmRevBlock.IB_DES

   AllRevInfo(35, 0) = "InitialsB_1"
   AllRevInfo(35, 1) = "3_Engineer"
   AllRevInfo(35, 2) = frmRevBlock.IB_ENG

   AllRevInfo(36, 0) = "InitialsB_1"
   AllRevInfo(36, 1) = "4_Check"
   AllRevInfo(36, 2) = frmRevBlock.IB_CHK

   AllRevInfo(37, 0) = "InitialsB_1"
   AllRevInfo(37, 1) = "5_Approved"
   AllRevInfo(37, 2) = frmRevBlock.IB_APP

   AllRevInfo(38, 0) = "InitialsC_1"
   AllRevInfo(38, 1) = "1_Drawn"
   AllRevInfo(38, 2) = frmRevBlock.IC_DRA

   AllRevInfo(39, 0) = "InitialsC_1"
   AllRevInfo(39, 1) = "2_Designer"
   AllRevInfo(39, 2) = frmRevBlock.IC_DES

   AllRevInfo(40, 0) = "InitialsC_1"
   AllRevInfo(40, 1) = "3_Engineer"
   AllRevInfo(40, 2) = frmRevBlock.IC_ENG

   AllRevInfo(41, 0) = "InitialsC_1"
   AllRevInfo(41, 1) = "4_Check"
   AllRevInfo(41, 2) = frmRevBlock.IC_CHK

   AllRevInfo(42, 0) = "InitialsC_1"
   AllRevInfo(42, 1) = "5_Approved"
   AllRevInfo(42, 2) = frmRevBlock.IC_APP

   For X = 0 To 43 Step 1

        modRevBlock.UpdateRevInfo AllRevInfo(X, 0), AllRevInfo(X, 1), AllRevInfo(X, 2)

   Next X

End Sub
```


REV
DATE
REVISION DESCRIPTION
DRA
DES
ENG
CHK
APP
0
Get Tags
Clear
Update
Close

```
Private Sub btnGetTitleInfo_Click()

   Dim GetInfo As TitleInfo

   GetInfo = modTitleBlock.GetTagInfo

   With frmTitleBlock

   .txtSubName = GetInfo.SubName
   .txtIndexNo = GetInfo.IndexNo
   .txtSheetNum = GetInfo.SheetNum
   .txtSheetOf = GetInfo.SheetOf
   .txtRevNum = GetInfo.RevNum
   .cmboxDistCode = GetInfo.DistCode
   .txtDescLine1 = GetInfo.DescLine1
   .txtDescLine2 = GetInfo.DescLine2
   .txtDescLine3 = GetInfo.DescLine3

   End With

End Sub

Private Sub btnUpdate_Click()

   Dim X As Long

   Dim AllTagInfo(9, 2) As String

   AllTagInfo(0, 0) = "SubstationName"
   AllTagInfo(0, 1) = "SubstationName"
   AllTagInfo(0, 2) = frmTitleBlock.txtSubName
   AllTagInfo(1, 0) = "SubstationIndexNo"
   AllTagInfo(1, 1) = "DocumentNo"
   AllTagInfo(1, 2) = frmTitleBlock.txtIndexNo
   AllTagInfo(2, 0) = "SheetNo"
   AllTagInfo(2, 1) = "Number"
   AllTagInfo(2, 2) = frmTitleBlock.txtSheetNum
   AllTagInfo(3, 0) = "SheetOf"
   AllTagInfo(3, 1) = "of #"
   AllTagInfo(3, 2) = frmTitleBlock.txtSheetOf
   AllTagInfo(4, 0) = "RevisionNo"
   AllTagInfo(4, 1) = "Number 0"
   AllTagInfo(4, 2) = frmTitleBlock.txtRevNum
   AllTagInfo(5, 0) = "DistributionCode"
   AllTagInfo(5, 1) = "Code"
   AllTagInfo(5, 2) = frmTitleBlock.cmboxDistCode.Value
   AllTagInfo(6, 0) = "DescriptionLines"
   AllTagInfo(6, 1) = "Line1"
   AllTagInfo(6, 2) = frmTitleBlock.txtDescLine1
   AllTagInfo(7, 0) = "DescriptionLines"
   AllTagInfo(7, 1) = "Line2"
   AllTagInfo(7, 2) = frmTitleBlock.txtDescLine2
   AllTagInfo(8, 0) = "DescriptionLines"
   AllTagInfo(8, 1) = "Line3"
   AllTagInfo(8, 2) = frmTitleBlock.txtDescLine3

   For X = 0 To 9 Step 1

        modTitleBlock.UpdateTagInfo AllTagInfo(X, 0), AllTagInfo(X, 1), AllTagInfo(X, 2)

   Next X

End Sub

Private Sub cmdClose_Click()
 Unload Me
End Sub

Private Sub TextBox7_Change()

End Sub

Private Sub txtDescLine1_Change()

End Sub

Private Sub txtIndexNo_Change()

End Sub

Private Sub txtRevNum_Change()

End Sub

Private Sub txtSubName_Change()

End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()

  frmTitleBlock.cmboxDistCode.AddItem "D-CON"
  frmTitleBlock.cmboxDistCode.AddItem "D-CHWRG"
  frmTitleBlock.cmboxDistCode.AddItem "D-LLAY"
  frmTitleBlock.cmboxDistCode.AddItem "D-ARRCG"
  frmTitleBlock.cmboxDistCode.AddItem "D-ONEL"
  frmTitleBlock.cmboxDistCode.AddItem "D-GLAY"
  frmTitleBlock.cmboxDistCode.AddItem "D-GF"
  frmTitleBlock.cmboxDistCode.AddItem "D-FSD"
  frmTitleBlock.cmboxDistCode.AddItem "D-OIL"
  'PUTS FIRST ITEM INTO TOP SPOT ON COMBO BOX
  frmTitleBlock.cmboxDistCode.ListIndex = 0

End Sub
```

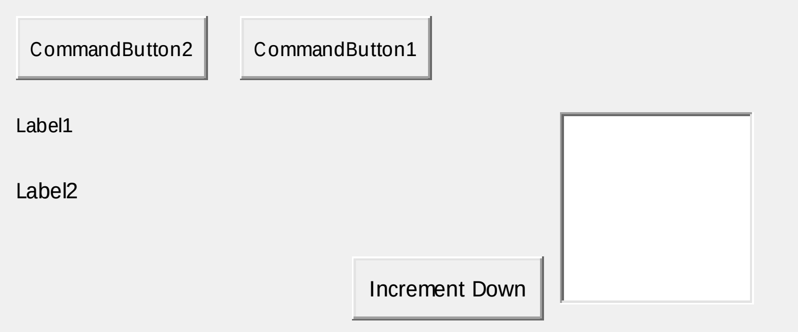


0

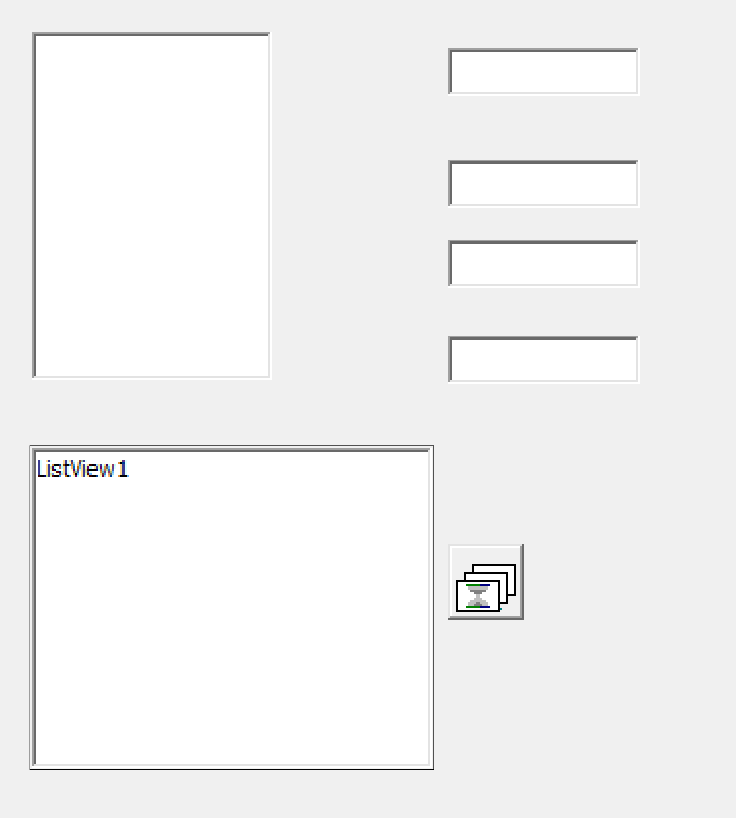
```
Private Sub CommandButton1_Click()

End Sub

Private Sub CommandButton1_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
  Call custom_KeyDown(KeyCode, Shift)
End Sub

Private Sub CommandButton2_Click()

End Sub

Private Sub CommandButton2_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
  Call custom_KeyDown(KeyCode, Shift)
End Sub

Private Sub custom_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
  Dim Icount As Integer
  Dim TooHigh As Integer
  Const TooLow = 0
  Const No_Selection = -1
  Const EnterKey = 13

  Select Case KeyCode
     Case EnterKey
       Label1.Caption = "Pressed ENTER"
     Case vbKeyF2
       Label1.Caption = "F2 pressed"
     Case vbKeyUp
       Label1.Caption = "Up arrow"
       Icount = ListBox1.ListIndex
       TooHigh = ListBox1.ListCount - 1

       'MsgBox str(icount)
        Select Case Icount
          Case No_Selection
            'Select first item
            ListBox1.ListIndex = 0
          Case TooLow
            'Wrap back to first item
            ListBox1.ListIndex = ListBox1.ListCount - 1
          Case Else
            ListBox1.ListIndex = Icount - 1
        End Select
    Case vbKeyDown
     Label1.Caption = "down arrow"
     Icount = ListBox1.ListIndex
     TooHigh = ListBox1.ListCount - 1
     'MsgBox str(icount)
        Select Case Icount
          Case No_Selection
            'Select first item
            ListBox1.ListIndex = 0
          Case TooHigh
```

```
            'Wrap back to first item
            ListBox1.ListIndex = 0
          Case Else
            ListBox1.ListIndex = Icount + 1
        End Select
   Case Else
    'do nothing
  End Select
End Sub

Private Sub IncrementDown_Click()

   Dim intCurSelect As Integer

   ListBox1.SetFocus
   If ListBox1.ListCount = 0 Then
     MsgBox "List Box is Empty!" & vbCrLf & "To fill box, Click -> LIST current DIR."
     Exit Sub
   End If

   If ListBox1.ListIndex = -1 Then
        'select the first in the list
        intCurSelect = 0
        'Label2.Caption = "nothing selected"

      Else
        intCurSelect = ListBox1.ListIndex
        Label2.Caption = intCurSelect

        'increment index and text to see if this is the
        'last item on the list, if so then set next to be the first item in list
        intCurSelect = intCurSelect + 1
        TooHigh = ListBox1.ListCount
        If intCurSelect = TooHigh Then
         intCurSelect = 0
        End If
   End If

        ListBox1.ListIndex = intCurSelect

        Label2.Caption = ListBox1.Value

   'open file


End Sub

Private Sub ListBox1_Click()
    Label1.Caption = "you clicked" & " " & ListBox1.Value & " listed at:" & Str(ListBox1.ListIndex)
End Sub

Private Sub ListBox1_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
  Call custom_KeyDown(KeyCode, Shift)
End Sub
```

```
Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
  ListBox1.AddItem "first"
  ListBox1.AddItem "second"
  ListBox1.AddItem "third"
  ListBox1.AddItem "fourth"
  ListBox1.AddItem "fifth"
  ListBox1.AddItem "sixth"
  ListBox1.AddItem "seventh"

End Sub
```


CommandButton2
CommandButton1
Label1
Label2
Increment Down

```
Private Sub ListView1_BeforeLabelEdit(Cancel As Integer)

End Sub

Private Sub TextBox1_KeyUp(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    Dim v2 As String

    TextBox2.Text = TextBox1.CurLine
    TextBox3.Text = TextBox1.CurX
    TextBox4.Text = TextBox1.CurTargetX
    TextBox5.Text = TextBox1.TabKeyBehavior

    v2 = KeyCode


    MsgBox v2




End Sub

Private Sub TextBox3_Change()

End Sub

Private Sub UserForm_Initialize()
    TextBox1.MultiLine = True

    TextBox1.Text = "Type your text here. User CTRL + ENTER to start a new line."
End Sub
```


ListView1

```
Sub TestColorTable()
  Dim MESSAGE As String
  MESSAGE = ColorTbl
  If MESSAGE <> "" Then
     MsgBox MESSAGE
    Else
     'do nothing
     MsgBox "found nothing"
  End If
End Sub

Function ColorTbl() As String

    Dim ct As ColorTable
    Dim ArrayRGBLongs() As Long
    Dim color As Long
    Dim r As Byte, g As Byte, b As Byte
    Dim MESSAGE As String

    MESSAGE = ""

    'Get a copy of the colortable that is currently attached to the active design file
    Set ct = Application.ActiveDesignFile.ExtractColorTable

    Dim cIndex As Long

    'Get an array of all of the RGB color values
    ArrayRGBLongs = ct.GetColors

    '35 is the highest color number assigned to a level
    For cIndex = 0 To 35

      ExtractRGB ArrayRGBLongs(cIndex), r, g, b

      MESSAGE = CheckColor(cIndex, r, g, b)

      If MESSAGE <> "" Then
        ColorTbl = MESSAGE & ". MORE COLORS MAY BE OFF! " & vbLf & "ATTACH CORRECT COLOR TABLE! "
        Exit Function
      End If
    Next
    ColorTbl = MESSAGE
    If MESSAGE <> "" Then
      MsgBox MESSAGE
    End If

End Function

Public Sub ExtractRGB(ByVal longColor As Long, intRed As Byte, intGreen As Byte, intBlue As Byte)
    Dim lngColor As Long

    lngColor = longColor
    intRed = lngColor Mod &H100
    lngColor = lngColor \ &H100
    intGreen = lngColor Mod &H100
    lngColor = lngColor \ &H100
    intBlue = lngColor Mod &H100
End Sub



Function CheckColor(chkIndex As Long, r As Byte, g As Byte, b As Byte) As String
 Dim MESSAGE As String
 Dim CheckIndex As Integer

 CheckIndex = CInt(chkIndex)

  MESSAGE = ""

      Select Case CheckIndex

' 0 is:  (r = 255, g = 255, b = 255)
        Case 0
            If r <> 255 Then
               MESSAGE = "Red for Color 0 should be 255 not:" & r & vbCr
            End If
            If g <> 255 Then
               MESSAGE = MESSAGE & "Green for color 0 not 255:" & g & vbCr
            End If
            If b <> 255 Then
               MESSAGE = MESSAGE & "Blue for color 0 should be 255 not: " & b & vbCr
            End If
               CheckColor = MESSAGE
            Exit Function

' 1 is:  (r = 0, g = 0, b = 255)
        Case 1
                If r <> 0 Then
               MESSAGE = "Red for color 1 should be 0 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 1 should be 0 not: " & g & vbCr
               End If
              If b <> 255 Then
               MESSAGE = MESSAGE & "Blue for color 1 should be 255 not: " & b & vbCr
            End If
                   CheckColor = MESSAGE
            Exit Function

' 2 is: (r = 0, g = 255, b = 0)
        Case 2
                If r <> 0 Then
               MESSAGE = "Red for color 2 should be 0 not: " & r & vbCr
               End If
              If g <> 255 Then
               MESSAGE = MESSAGE & "Green for color 2 should be 255 not: " & g & vbCr
               End If
              If b <> 0 Then
```

```
               MESSAGE = MESSAGE & "Blue for color 2 should be 0 not: " & b & vbCr
            End If
                  CheckColor = MESSAGE
            Exit Function

' 3 is:  (r = 255, g = 0, b = 0)
        Case 3
                If r <> 255 Then
               MESSAGE = "Red for color 3 should be 255 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 3 should be 0 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 3 should be 0 not: " & b & vbCr
            End If
                 CheckColor = MESSAGE
            Exit Function

' 4 is:  (r = 255, g = 255, b = 0)
        Case 4
                If r <> 255 Then
               MESSAGE = "Red for color 4 should be 255 not: " & r & vbCr
               End If
              If g <> 255 Then
               MESSAGE = MESSAGE & "Green for color 4 should be 255 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 4 should be 0 not: " & b & vbCr
            End If
                  CheckColor = MESSAGE
            Exit Function

' 5 is:  (r = 255, g = 0, b = 255)
        Case 5
                If r <> 255 Then
               MESSAGE = "Red for color 5 should be 255 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 5 should be 0 not: " & g & vbCr
               End If
              If b <> 255 Then
               MESSAGE = MESSAGE & "Blue for color 5 should be 255 not: " & b & vbCr
            End If
                CheckColor = MESSAGE
            Exit Function

' 6 is: (r = 255, g = 127, b = 0)
        Case 6
                If r <> 255 Then
               MESSAGE = "Red for color 6 should be 255 not: " & r & vbCr
               End If
              If g <> 127 Then
               MESSAGE = MESSAGE & "Green for color 6 should be 127 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 6 should be 0 not: " & b & vbCr
            End If
               CheckColor = MESSAGE
            Exit Function

' 7 is:  (r = 0, g = 255, b = 255)
        Case 7
                If r <> 0 Then
               MESSAGE = "Red for color 7 should be 0 not: " & r & vbCr
               End If
              If g <> 255 Then
               MESSAGE = MESSAGE & "Green for color 7 should be 255 not: " & g & vbCr
               End If
              If b <> 255 Then
               MESSAGE = MESSAGE & "Blue for color 7 should be 255 not: " & b & vbCr
            End If
               CheckColor = MESSAGE
            Exit Function

' 8 is:  (r = 64, g = 64, b = 64)
        Case 8
                If r <> 64 Then
               MESSAGE = "Red for color 8 should be 64 not: " & r & vbCr
               End If
              If g <> 64 Then
               MESSAGE = MESSAGE & "Green for color 8 should be 64 not: " & g & vbCr
               End If
              If b <> 64 Then
               MESSAGE = MESSAGE & "Blue for color 8 should be 64 not: " & b & vbCr
            End If
               CheckColor = MESSAGE
            Exit Function

' 9 is:  (r = 192, g = 192, b = 192)
        Case 9
                If r <> 192 Then
               MESSAGE = "Red for color 9 should be 192 not: " & r & vbCr
               End If
              If g <> 192 Then
               MESSAGE = MESSAGE & "Green for color 9 should be 192 not: " & g & vbCr
               End If
              If b <> 192 Then
               MESSAGE = MESSAGE & "Blue for color 9 should be 192 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function

' 10 is:  (r = 254, g = 0, b = 96)

        Case 10
                If r <> 254 Then
               MESSAGE = "Red for color 10 should be 254 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 10 should be 0 not: " & g & vbCr
               End If
              If b <> 96 Then
               MESSAGE = MESSAGE & "Blue for color 10 should be 96 not: " & b & vbCr
            End If
             CheckColor = MESSAGE
            Exit Function
' 11 is:  (r = 160, g = 224, b = 0)
        Case 11
                If r <> 160 Then
               MESSAGE = "Red for color 11 should be 160 not: " & r & vbCr
               End If
              If g <> 224 Then
               MESSAGE = MESSAGE & "Green for color 11 should be 224 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 11 should be 0 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function

' 12 is:  (r = 0, g = 254, b = 160)
        Case 12
                If r <> 0 Then
               MESSAGE = "Red for color 12 should be 0 not: " & r & vbCr
               End If
              If g <> 254 Then
               MESSAGE = MESSAGE & "Green for color 12 should be 254 not: " & g & vbCr
               End If
              If b <> 160 Then
               MESSAGE = MESSAGE & "Blue for color 12 should be 160 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function

' 13 is:  (r = 128, g = 0, b = 160)
        Case 13
                If r <> 128 Then
               MESSAGE = "Red for color 13 should be 128 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 13 should be 0 not: " & g & vbCr
               End If
              If b <> 160 Then
               MESSAGE = MESSAGE & "Blue for color 13 should be 160 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function

' 14 is:  (r = 176, g = 176, b = 176)
        Case 14
                If r <> 176 Then
               MESSAGE = "Red for color 14 should be 176 not: " & r & vbCr
               End If
              If g <> 176 Then
               MESSAGE = MESSAGE & "Green for color 14 should be 176 not: " & g & vbCr
               End If
              If b <> 176 Then
               MESSAGE = MESSAGE & "Blue for color 14 should be 176 not: " & b & vbCr
            End If
                 CheckColor = MESSAGE
            Exit Function

' 15 is:  (r = 0, g = 240, b = 240)
        Case 15
                If r <> 0 Then
               MESSAGE = "Red for color 15 should be 0 not: " & r & vbCr
               End If
              If g <> 240 Then
               MESSAGE = MESSAGE & "Green for color 15 should be 240 not: " & g & vbCr
               End If
              If b <> 240 Then
               MESSAGE = MESSAGE & "Blue for color 15 should be 240 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function

' 16 is:  (r = 240, g = 240, b = 240)
        Case 16
              If r <> 240 Then
               MESSAGE = "Red for color 16 should be 240 not: " & r & vbCr
               End If
              If g <> 240 Then
               MESSAGE = MESSAGE & "Green for color 16 should be 240 not: " & g & vbCr
               End If
              If b <> 240 Then
               MESSAGE = MESSAGE & "Blue for color 16 should be 240 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function

' 17 is:  (r = 0, g = 0, b = 240)
        Case 17
                If r <> 0 Then
               MESSAGE = "Red for color 17 should be 0 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 17 should be 0 not: " & g & vbCr
               End If
              If b <> 240 Then
               MESSAGE = MESSAGE & "Blue for color 17 should be 240 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function
```

```
' 18 is:  (r = 0, g = 240, b = 0)
        Case 18
                If r <> 0 Then
               MESSAGE = "Red for color 18 should be 0 not: " & r & vbCr
               End If
              If g <> 240 Then
               MESSAGE = MESSAGE & "Green for color 18 should be 240 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 18 should be 0 not: " & b & vbCr
            End If
             CheckColor = MESSAGE
            Exit Function


' 19 is:  (r = 240, g = 0, b = 0)
        Case 19
                If r <> 240 Then
               MESSAGE = "Red for color 19 should be 240 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 19 should be 0 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 19 should be 0 not: " & b & vbCr
            End If
              CheckColor = MESSAGE
            Exit Function

' 20 is:  (r = 240, g = 240, b = 0)

        Case 20
                If r <> 240 Then
               MESSAGE = "Red for color 20 should be 240 not: " & r & vbCr
               End If
              If g <> 240 Then
               MESSAGE = MESSAGE & "Green for color 20 should be 240 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 20 should be 0 not: " & b & vbCr
            End If
             CheckColor = MESSAGE
            Exit Function

' 21 is:  (r = 240, g = 0, b = 240)
        Case 21
                If r <> 240 Then
               MESSAGE = "Red for color 21 should be 240 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 21 should be 0 not: " & g & vbCr
               End If
              If b <> 240 Then
               MESSAGE = MESSAGE & "Blue for color 21 should be 240 not: " & b & vbCr
            End If
             CheckColor = MESSAGE
            Exit Function

' 22 is:  (r = 240, g = 122, b = 0)
        Case 22
                If r <> 240 Then
               MESSAGE = "Red for color 22 should be 240 not: " & r & vbCr
               End If
              If g <> 122 Then
               MESSAGE = MESSAGE & "Green for color 22 should be 122 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 22 should be 0 not: " & b & vbCr
            End If
             CheckColor = MESSAGE
            Exit Function

' 35 is: r = 225, g = 0, b = 0)
        Case 35
                If r <> 225 Then
               MESSAGE = "Red for color 35 should be 225 not: " & r & vbCr
               End If
              If g <> 0 Then
               MESSAGE = MESSAGE & "Green for color 35 should be 0 not: " & g & vbCr
               End If
              If b <> 0 Then
               MESSAGE = MESSAGE & "Blue for color 35 should be 0 not: " & b & vbCr
            End If
             CheckColor = MESSAGE
            Exit Function

        Case Else
              CheckColor = ""
        End Select

End Function
```

```
Option Explicit
Option Base 1
Const Brd_D10 As String = "BDR-D10"
Const Brd_E10 As String = "BDR-E10"
Const Brd_D12 As String = "BDR-D12"
Const Brd_E12 As String = "BDR-E12"
Const Brd_T10 As String = "BDR-T10"
Const Brd_T12 As String = "BDR-T12"
'***********************************


Sub F2()
    'by Keith Knowles 12/10/2013

    '*********declarations **********************
    Dim msg As String
    msg = ""
    Dim rMsg As String
    rMsg = ""
    Dim bdrElement As CellElement
    Dim T_BorderIgnore As Boolean
    T_BorderIgnore = False 'True puts fence around T borders - needed for final prints
    Dim BorderExists As Boolean
    BorderExists = True  'assume a border exists
    Dim FenceGood As Boolean
    Dim varTestString As Variant
    Dim ScaledDwgFlag As Boolean
    ScaledDwgFlag = False


    '******* Get Getborder and DrawFence *********
    Set bdrElement = GetBorder(T_BorderIgnore, BorderExists)
    'if no border then skip over DrawFence
        If BorderExists = True Then
            'find out if border is a scaled border
            varTestString = InStr(1, bdrElement.Name, "10", vbTextCompare)
            If varTestString = 0 Then
                'do nothing
              Else
                'scaled dwg - set scaledDwgFlag to true
                 ScaledDwgFlag = True
             End If

            'draw the fence

             FenceGood = DrawFence(bdrElement, msg)
             rMsg = rMsg & msg
                If FenceGood = True Then
                  'do nothing
                End If
          Else
            'do nothing
             rMsg = rMsg & "No valid border! FENCE NOT DRAWN! "
```

```
         End If
     'Clear object variable
      Set bdrElement = Nothing


   '**********Get Resolution and Set General Text Height***********
   If BorderExists = True Then
        Dim NoErrorVal As Boolean
        NoErrorVal = True
        msg = GetResolSetGenTextHgt(NoErrorVal)
        rMsg = rMsg & msg
   End If



   '*******Zoom into the Title Block Area*********
    If BorderExists = True Then
        If FenceGood = True Then
            GetRange  'zoom into Title Block Area
        End If
    End If


    '********Check Color Table********************
      msg = ""
      msg = ColorTable_01.ColorTbl

      If msg <> "" Then
             rMsg = rMsg & " Color Table Error! "
      End If


    '******Check for Levels***************************
     msg = ""
     msg = Levels.CheckLevels()
     If msg = "" Then
        'do nothing levels are fine
        Else
         rMsg = rMsg & msg
     End If
    '*******END OF  Check for Levels********************



    '***********raster attached? ***********************
    msg = ""
    msg = RasterAttached
    rMsg = rMsg & msg


    '**********Scan for Dimensions*****************
    'explode dimensions with "Graphic Group" ON so that
    'dim values do not change when the working resolution
    'is changed
```

```
      If ScaledDwgFlag = False Then
        'only check for Dimensions in Scaled dwgs
        msg = ""
        msg = ScanForDims
        rMsg = rMsg & msg
      End If


    '************fit view 5 to extents*****************
    If ActiveDesignFile.Views(5).IsOpen Then
        CadInputQueue.SendCommand "FIT VIEW EXTENDED 5"
      Else
        rMsg = rMsg & "FIX VIEW 5! "
    End If

    '***********reset command to selection***********
        CadInputQueue.SendCommand "CHOOSE ELEMENT"
        CommandState.StartDefaultCommand


    '************SHOW ISSUES TO USERS******************
    'priority 10 red, 11 yellow, 12 info, 14 none
    If rMsg = "" Then
        ShowTempMessage msdStatusBarAreaLeft, "Nothing to Report!"
        ShowTempMessage Area:=msdStatusBarAreaMiddle, MESSAGE:="DRAWING CHECKS OK! "
        Exit Sub
    End If

        'Set lock settings: Snap on, Unit on, Graphic Group on, Axis lock off, Grid lock off
        FileAttributeSettings

    If rMsg <> "" Then
        ShowTempMessage msdStatusBarAreaLeft, "ERRORS or Information:"
        ShowTempMessage Area:=msdStatusBarAreaMiddle, MESSAGE:=rMsg
        MessageCenter.AddMessage MESSAGE:=rMsg, Priority:=11, openalertdialog:=False
    End If

End Sub
Function DrawFence(BDR As CellElement, retMsg As String) As Boolean
     Dim delta_Y As Variant
     Dim delta_X As Variant
     Dim D12_Ratio As Double
     D12_Ratio = 1.54545454545455
     Const E12_Ratio As Double = 1.4
     Dim FortyTwo As Variant
     FortyTwo = 42#
     Dim ThirtyFour As Variant
     ThirtyFour = 34#
     Dim curElem As Element
     Dim lngScaleX As Double
     Dim lngScaleY As Double

     DrawFence = True       'predict success
     retMsg = ""            'predict success
```

```
        'BDR.Origin.x
     Dim pts(1 To 4) As Point3d

        'non-scaled drawings
      If BDR.Name = Brd_D10 Then
        pts(1).X = BDR.Range.Low.X
        pts(1).Y = BDR.Range.Low.Y
        pts(2).X = BDR.Range.Low.X + ThirtyFour
        pts(2).Y = BDR.Range.Low.Y
        pts(3).X = BDR.Range.Low.X + ThirtyFour
        pts(3).Y = BDR.Range.High.Y
        pts(4).X = BDR.Range.Low.X
        pts(4).Y = BDR.Range.High.Y

       ElseIf BDR.Name = Brd_E10 Then
        pts(1).X = BDR.Range.Low.X
        pts(1).Y = BDR.Range.Low.Y
        pts(2).X = BDR.Range.Low.X + FortyTwo
        pts(2).Y = BDR.Range.Low.Y
        pts(3).X = BDR.Range.Low.X + FortyTwo
        pts(3).Y = BDR.Range.High.Y
        pts(4).X = BDR.Range.Low.X
        pts(4).Y = BDR.Range.High.Y


       ElseIf BDR.Name = Brd_T10 Or BDR.Name = Brd_T12 Then
        pts(1).X = BDR.Range.Low.X
        pts(1).Y = BDR.Range.Low.Y
        pts(2).X = BDR.Range.High.X
        pts(2).Y = BDR.Range.Low.Y
        pts(3).X = BDR.Range.High.X
        pts(3).Y = BDR.Range.High.Y
        pts(4).X = BDR.Range.Low.X
        pts(4).Y = BDR.Range.High.Y

        'scaled drawings
       ElseIf BDR.Name = Brd_D12 Then
        delta_Y = BDR.Range.High.Y - BDR.Range.Low.Y
        'deduce delta_X by  Ratio 17/11
        delta_X = D12_Ratio * delta_Y
        pts(1).X = BDR.Range.Low.X
        pts(1).Y = BDR.Range.Low.Y
        pts(2).X = BDR.Range.Low.X + delta_X
        pts(2).Y = BDR.Range.Low.Y
        pts(3).X = BDR.Range.Low.X + delta_X
        pts(3).Y = BDR.Range.High.Y
        pts(4).X = BDR.Range.Low.X
        pts(4).Y = BDR.Range.High.Y

       ElseIf BDR.Name = Brd_E12 Then
        delta_Y = BDR.Range.High.Y - BDR.Range.Low.Y
        'deduce delta_X by ratio 14/10
        delta_X = E12_Ratio * delta_Y
        pts(1).X = BDR.Range.Low.X
        pts(1).Y = BDR.Range.Low.Y
        pts(2).X = BDR.Range.Low.X + delta_X
        pts(2).Y = BDR.Range.Low.Y
        pts(3).X = BDR.Range.Low.X + delta_X
        pts(3).Y = BDR.Range.High.Y
        pts(4).X = BDR.Range.Low.X
        pts(4).Y = BDR.Range.High.Y
       Else
         'Big Problem -- no border to draw a fence around
         retMsg = "NO BORDER FOUND! "
         DrawFence = False
         Exit Function
     End If

    ' Create the fence
    With ActiveDesignFile.Fence
       .DefineFromModelPoints 1, pts()
       .Draw msdDrawingModeHilite
    End With

    If BDR.Range.Low.X >= 0.1 Or BDR.Range.Low.Y >= 0.1 Then
    'If BDR.Range.Low.x <> 0 Or BDR.Range.Low.Y <> 0 Then

      retMsg = "BORDER OFF 0,0! "
    End If

End Function
Function GetBorder(ignoreT As Boolean, BorderExist As Boolean) As Element

  Dim number As Double


  Dim rngBDR As Range3d
  Dim pntBDRs As Point3d
  Dim pntBDRe As Point3d
  Dim rngTBDR As Range3d
  Dim pntTBDRs As Point3d
  Dim pntTBDRe As Point3d
  Dim dblScale As Double

  Dim BorderName As String
  Dim oElem As Element
  Dim oCellElem As CellElement
  Dim BdrObject As CellElement
  Dim TbdrObject As CellElement
  Dim oEnum As ElementEnumerator
  Dim ElementCounter As Long
  Dim BorderType As String

  Dim BorderD10 As Boolean
  Dim BorderE10 As Boolean
  Dim BorderT10 As Boolean
  Dim BorderD12 As Boolean
  Dim BorderE12 As Boolean
  Dim BorderT12 As Boolean

  BorderD10 = False
  BorderE10 = False
  BorderT10 = False
  BorderD12 = False
  BorderE12 = False
  BorderT12 = False

  Set oEnum = ActiveModelReference.Scan()

   While oEnum.MoveNext
        ElementCounter = ElementCounter + 1
        Set oElem = oEnum.Current

        If oElem.IsCellElement Then
          Set oCellElem = oElem

          Select Case oCellElem.Name
            Case Brd_D10
              'MsgBox "D10"
              Set BdrObject = oCellElem
              BorderD10 = True
            Case Brd_D12
              'MsgBox "D12"
              Set BdrObject = oCellElem
              BorderD12 = True
            Case Brd_E10
              'MsgBox "E10"
              Set BdrObject = oCellElem
              BorderE10 = True
            Case Brd_E12
              'MsgBox "E12"
              Set BdrObject = oCellElem
              BorderE12 = True
            Case Brd_T10
              'MsgBox "T10"
              Set TbdrObject = oCellElem
              BorderT10 = True
            Case Brd_T12
              'MsgBox "T12"
              Set TbdrObject = oCellElem
              BorderT12 = True
            Case Else
              'No border found - send by reference this value to avoid error
              BorderExist = False
           End Select
        End If
   Wend

   If ignoreT = True Then
     'MsgBox "ignore T"
```

```
       If BorderE10 Or BorderD10 Or BorderD12 Or BorderE12 Then
          Set GetBorder = BdrObject
          BorderExist = True
          Exit Function
         Else
          'do nothing
          'MsgBox "No D or E Borders in this file"
          BorderExist = False
       End If
   End If

   If ignoreT = False Then
       If BorderT10 Or BorderT12 Then
          BorderExist = True
          Set GetBorder = TbdrObject
          'MsgBox "T border takes priority"
         ElseIf BorderE10 Or BorderD10 Or BorderD12 Or BorderE12 Then
          BorderExist = True
          Set GetBorder = BdrObject
          Exit Function
         Else
          'MsgBox "No D or E Borders in this file"
          BorderExist = False
       End If
   End If
End Function
Sub TEST_RasterAttached()

  Dim rtnMsg As String
  rtnMsg = RasterAttached
  MsgBox rtnMsg

End Sub
Function RasterAttached() As String

  Dim RtrnMsg As String
  Dim RasterFullName As String
  Dim RasterPath As String
  Dim RasterFileSize As Long
  Dim DesignFileName As String
  Dim DesignFilePath As String
  Dim Icount As Integer
  Dim intAttachCount As Integer

  Dim strAttachName As String
  Dim strAttachPath As String

  Dim strFilename As String
  Dim strFilepath As String

  Dim CurrentFile As DesignFile
  Set CurrentFile = Application.ActiveDesignFile

  strFilename = CurrentFile.Name
  strFilepath = CurrentFile.path

  Dim RasterCount As Integer
        'gets the name of the first attached raster and the number of rasters attached
        'need full path since raster could be in another folder
   Dim att As Rasters
   Set att = Application.RasterManager.Rasters
        'are there any attachments

        If att.count = 0 Then
              'get path
                 If Right(strFilepath, 1) <> "\" Then
                   strFilepath = strFilepath & "\"
                 End If
              'get filename & remove ".dgn"
                 strFilename = CurrentFile.Name
              'take off .dgn filename and add .cit
                 Dim Pos1 As Variant
                 Dim Fullpath As String
                 Pos1 = InStr(1, strFilename, ".", vbTextCompare)
                 strFilename = Left(strFilename, Pos1)
                 strFilename = strFilename & "cit"
               'check the folder for the *.cit file
                 Fullpath = strFilepath & strFilename
                 Dim fname As String
                 fname = Dir(Fullpath)
                 If fname <> "" Then
                     RasterAttached = "Raster Detached: " & fname
                     Exit Function
                   Else
                     RasterAttached = ""
                     Exit Function
                 End If

           Else
              strAttachPath = att.Item(1).RasterInformation.path
              strAttachName = att.Item(1).RasterInformation.Name
              'RasterFileSize = att.Item(1).RasterInformation.FileSize
              ' MsgBox Str(RasterFileSize)
        End If

  RtrnMsg = ""  'keep track of all issues
  'Test Path Name
        If StrComp(strAttachPath, strFilepath, vbTextCompare) = 0 Then
            'path match
          Else
            RtrnMsg = "Raster Path Different! " & vbLf
        End If

  'Test File Name
        'take of .cit and .dgn filename
        Dim Posit1 As Variant
        Dim Posit2 As Variant
        Posit1 = InStr(1, strAttachName, ".", vbTextCompare)
        strAttachName = Left(strAttachName, Posit1)
        Posit2 = InStr(1, strFilename, ".", vbTextCompare)
        strFilename = Left(strFilename, Posit2)
        If StrComp(strAttachName, strFilename, vbTextCompare) = 0 Then
             'name match
          Else
             RtrnMsg = RtrnMsg & "RASTER NAMING ERROR! " & vbLf
        End If

  'Test for too many rasters attached
        If att.count >= 2 Then
        RtrnMsg = RtrnMsg & "TOO MANY RASTERS! "
        End If

   RasterAttached = RtrnMsg

End Function



Sub FileAttributeSettings()
  Dim MESSAGE As String
  Dim SnapE As Boolean
  Dim UnitL As Boolean
  Dim graphG As Boolean
  Dim activeR As Boolean
  Dim ActRefMod As ModelReference

  With Application.ActiveSettings
      .SnapLockEnabled = True
      .UnitLockEnabled = True
      .GraphicGroupLockEnabled = True
        '    .GridUnits
        '    .GridReference
       .AxisLockEnabled = False
       .GridLockEnabled = False

  End With

  CadInputQueue.SendKeyin "LOCK SNAP KEYpoint"

' If activeR = Application.HasActiveModelReference Then
'   Set ActRefMod = Application.ActiveModelReference
' End If

'   MsgBox SnapE & UnitL

End Sub



Function ScanForDims() As String
    Dim counter As Integer
    Dim myElement As Element
    Dim myEnum As ElementEnumerator
    Dim myDim As DimensionElement

    Dim esc As ElementScanCriteria
    Set esc = New ElementScanCriteria

    Dim getDimText As ComplexElement

    esc.ExcludeAllTypes
    esc.IncludeType msdElementTypeDimension

    Set myEnum = ActiveModelReference.Scan(esc)
    counter = 0

    Do While myEnum.MoveNext

        Set myDim = myEnum.Current



        If myDim.IsTextElement Then
          'MsgBox myDim.IsTextElement
        End If

         If myDim.IsGraphical Then
          'MsgBox myDim.IsGraphical
          'MsgBox myDim.ActualValue
          'Set getDimText = myDim.AsComplexElement

          'getDimText.GetSubElements


        End If

        'MsgBox Str(dimValue) & "zeros " & Str(ZeroValue)
        counter = counter + 1

    Loop

    If counter > 0 Then
       ScanForDims = "(" & Str(counter) & " Dimensions Found!) "
       Exit Function
    End If
     ScanForDims = ""
End Function

Sub test_ScanForDims()

   Dim msg As String
   msg = ScanForDims
   MsgBox msg

End Sub
```

```
Sub GetRange()
 On Error GoTo errhnd
 Dim lngDspPrty As Long

 Dim ele As CellElement
 Dim success As Boolean
 success = False
 Dim rng As Range3d
 Dim BorderName As String

 Set ele = GetBorder(False, True)
 rng = ele.Range
 BorderName = ele.Name

 success = ZoomToTitle(rng, BorderName, 1)

 'MsgBox success
errhnd:
   Select Case Err.number
     Case 91 'Get Border didn't find any Borders
             'Could be a raster file a raster Title Block
       'MsgBox "Program ended! No title block on this drawing."
       Err.Clear
     End Select
End Sub

Function ZoomToTitle(Rngr As Range3d, BDR_X1X As String, viewNmbr As Integer) As Boolean
 Dim dblFactor As Double
 Dim DeltaY As Double
 Dim DeltaX As Double
 Dim oView As View
 Set oView = ActiveDesignFile.Views(viewNmbr)
 Dim pntOrigin As Point3d
 Dim rngExtents As Range3d
 Dim pntExtents As Point3d
 Dim myLine As LineElement
 Dim pntZoom As Point3d

 'Establish extents just around the Title Block area
 'This allows for extra elements outside of the Title Block
 'area to not affect the zoom into the title area of the Title
 'Block
 '*******************************************
 rngExtents = Rngr

 oView.Origin = rngExtents.Low

 pntExtents.X = rngExtents.High.X - rngExtents.Low.X
 pntExtents.Y = rngExtents.High.Y - rngExtents.Low.Y

 oView.Extents = pntExtents
 'oView.Redraw
 'oView.Redraw
 '********************************************
 pntOrigin.X = Rngr.Low.X
 pntOrigin.Y = Rngr.Low.Y
 pntOrigin.Z = 0

 DeltaX = Rngr.High.X - Rngr.Low.X
 DeltaY = Rngr.High.Y - Rngr.Low.Y

 Select Case BDR_X1X
    Case Brd_D10
        With Rngr
          pntZoom.X = .Low.X + (1.488623 * DeltaY)
          pntZoom.Y = .Low.Y + (0.2227318 * DeltaY)
        End With
        dblFactor = 0.45
     Case Brd_E10
        With Rngr
          pntZoom.X = .Low.X + (1.358326 * DeltaY)
          pntZoom.Y = .Low.Y + (0.163336 * DeltaY)
        End With
        dblFactor = 0.33
     Case Brd_D12
        With Rngr
          pntZoom.X = .Low.X + (1.488623 * DeltaY)
          pntZoom.Y = .Low.Y + (0.2227318 * DeltaY)
        End With
        dblFactor = 0.45
     Case Brd_E12
        With Rngr
          pntZoom.X = .Low.X + (1.358323 * DeltaY)
          pntZoom.Y = .Low.Y + (0.163336 * DeltaY)
        End With
        dblFactor = 0.33
     Case Brd_T10
        With Rngr
          pntZoom.X = .Low.X + (0.9702261904 * DeltaX)
          pntZoom.Y = .Low.Y + (0.16335 * DeltaY)
        End With
        dblFactor = 0.33
     Case Brd_T12
        With Rngr
          pntZoom.X = .Low.X + (0.9702214 * DeltaX)
          pntZoom.Y = .Low.Y + (0.16333 * DeltaY)
        End With
        dblFactor = 0.33
     Case Else
       ZoomToTitle = False
       Exit Function
 End Select


'      With Application
'        Set myLine = .CreateLineElement2(Nothing, pntOrigin, pntZoom)
'        .ActiveModelReference.AddElement myLine
'      End With


 'Zoom about the center of the range.
 oView.ZoomAboutPoint pntZoom, dblFactor
 oView.Redraw
 oView.Redraw
 ZoomToTitle = True


End Function

Sub test_GetResolSetGenTextHgt()
    Dim msg As String
    msg = GetResolSetGenTextHgt
    MsgBox msg
End Sub

Function GetResolSetGenTextHgt(NoError As Boolean) As String
 On Error GoTo errhnd
 Dim lngDspPrty As Long
 Dim retMsg As String
 Dim Resl As String
 Resl = ""
 Dim ele As CellElement
 Dim rng As Range3d
 Dim BorderName As String
 Dim msg As String
 msg = ""
 Set ele = GetBorder(True, True)
 rng = ele.Range
 BorderName = ele.Name

 '****************************************
 Dim DeltaY As Long
 DeltaY = rng.High.Y - rng.Low.Y

 Select Case BorderName
     Case Brd_D10
         'if deltaY   - check if 10000
         If DeltaY <> 22 Then
            msg = "BORDER SIZE OR RESOLUTION ERROR! "
           Else
            Resl = "10000"
         End If
     Case Brd_E10
         'if deltaY   - check if 10000
         If DeltaY <> 30 Then
            msg = "BORDER SIZE OR RESOLUTION ERROR! "
           Else
            Resl = "10000"
         End If
     Case Brd_D12
        'if deltaY is not when one of the following
        'send out working resolution error
        Select Case DeltaY
          Case 352
             '1/16"=1'-0"
             Resl = "750"
          Case 235
             '3/32"=1'-0"
             Resl = "1125"
          Case 176
             '1/8"=1'-0"
             Resl = "1500"
          Case 117
             '3/16"=1'-0"
             Resl = "2250"
          Case 88
             '1/4"=1'-0"
             Resl = "3000"
          Case 59
             '3/8"=1'-0"
             Resl = "4500"
          Case 44
             '1/2"=1'-0"
             Resl = "6000"
          Case 29
             '3/4"=1'-0"
             Resl = "9000"
          Case 22
             '1"=1'-0"
             Resl = "12000"
          Case 15
             '1 1/2"=1'-0"
             Resl = "18000"
          Case 11
             '2"=1'-0"
             Resl = "24000"
          Case 7
             '3"=1'-0"
             Resl = "36000"
          Case Else
             'Working Resolution is incorrect
             msg = "BORDER SIZE OR RESOLUTION ERROR! "
          End Select

     Case Brd_E12
          'if deltaY is not when one of the following
          'send out working resolution error
           Select Case DeltaY
          Case 480
             '1/16"=1'-0"
             Resl = "750"
          Case 320
             '3/32"=1'-0"
             Resl = "1125"
          Case 240
```

```
             '1/8"=1'-0"
             Resl = "1500"
          Case 160
             '3/16"=1'-0"
             Resl = "2250"
          Case 120
             '1/4"=1'-0"
             Resl = "3000"
          Case 80
             '3/8"=1'-0"
             Resl = "4500"
          Case 60
             '1/2"=1'-0"
             Resl = "6000"
          Case 40
             '3/4"=1'-0"
             Resl = "9000"
          Case 30
             '1"=1'-0"
             Resl = "12000"
          Case 20
             '1 1/2"=1'-0"
             Resl = "18000"
          Case 15
             '2"=1'-0"
             Resl = "24000"
          Case 10
             '3"=1'-0"
             Resl = "36000"
          Case Else
             'Working Resolution is incorrect
             msg = "BORDER SIZE OR RESOLUTION ERROR! "
          End Select
     Case Else
       'Not expecting to have no Border, but Cell name could have been changed
       msg = "(Border Cell Name is Unexpected!) "
End Select

Select Case Resl
        'Set General Text Height according drawing scale
          Case "10000"
             'Non-Scaled
             ActiveSettings.TextStyle.Height = 0.1
             ActiveSettings.TextStyle.Width = 0.1
          Case "750"
             ActiveSettings.TextStyle.Height = 1.60026
             ActiveSettings.TextStyle.Width = 1.60026
          Case "1125"
             ActiveSettings.TextStyle.Height = 1.06640625
             ActiveSettings.TextStyle.Width = 1.06640625
          Case "1500"
             ActiveSettings.TextStyle.Height = 0.799479
             ActiveSettings.TextStyle.Width = 0.799479
          Case "2250"
```

```
             ActiveSettings.TextStyle.Height = 0.533854
             ActiveSettings.TextStyle.Width = 0.533854
          Case "3000"
             ActiveSettings.TextStyle.Height = 0.399739
             ActiveSettings.TextStyle.Width = 0.399739
          Case "4500"
             ActiveSettings.TextStyle.Height = 0.266927
             ActiveSettings.TextStyle.Width = 0.266927
          Case "6000"
             ActiveSettings.TextStyle.Height = 0.200521
             ActiveSettings.TextStyle.Width = 0.200521
          Case "9000"
             ActiveSettings.TextStyle.Height = 0.1328125
             ActiveSettings.TextStyle.Width = 0.1328125
          Case "12000"
             ActiveSettings.TextStyle.Height = 0.100260416
             ActiveSettings.TextStyle.Width = 0.100260416
          Case "18000"
             ActiveSettings.TextStyle.Height = 0.06640625
             ActiveSettings.TextStyle.Width = 0.06640625
          Case "24000"
             ActiveSettings.TextStyle.Height = 0.0494792
             ActiveSettings.TextStyle.Width = 0.0494792
          Case "36000"
             ActiveSettings.TextStyle.Height = 0.03385416
             ActiveSettings.TextStyle.Width = 0.03385416
          Case Else
             'do nothing
 End Select

  If msg = "" Then
     retMsg = "(dwg resolution: " & Resl & ") "
     GetResolSetGenTextHgt = retMsg
     NoError = True

    Else
     GetResolSetGenTextHgt = msg
     NoError = False
    End If

  Exit Function
errhnd:
   Select Case Err.number
     Case 91 'Get Border didn't find any Borders
             'Could be a raster file a raster Title Block
             'MsgBox "Program ended! No title block on this drawing."
       Err.Clear
     End Select
     GetResolSetGenTextHgt = "(No Border or Border Cell Name Changed!) "
End Function
```

```
Sub FileAttributeSettings()
  Dim MESSAGE As String
  Dim SnapE As Boolean
  Dim UnitL As Boolean
  Dim graphG As Boolean
  Dim activeR As Boolean
  Dim ActRefMod As ModelReference

  With Application.ActiveSettings
      .Level.Name
      .ColorName
      .LineWeight
      .LineStyle

      .SnapLockEnabled = True
      .UnitLockEnabled = True
      .GraphicGroupLockEnabled = True
        '    .GridUnits
        '    .GridReference
       .AxisLockEnabled = False
       .GridLockEnabled = False

  End With

  CadInputQueue.SendKeyin "LOCK SNAP KEYpoint"

' If activeR = Application.HasActiveModelReference Then
'   Set ActRefMod = Application.ActiveModelReference
' End If

'   MsgBox SnapE & UnitL

End Sub


Private Sub cmbCloudEl_Click()
 ' MsgBox "hello world"
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Backcircle"""

    CadInputQueue.SendCommand "ACTIVE WEIGHT 3"
'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 1, "MGDSHOOK"


'   Start a command
    CadInputQueue.SendCommand "PLACE REVCLOUD ELEMENT"

'    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * 0.1), "COMPCURV"
```

```
End Sub

Private Sub cmbCloudPt_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Backcircle"""

    CadInputQueue.SendCommand "ACTIVE WEIGHT 3"

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 1, "MGDSHOOK"

'    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * 0.1), "COMPCURV"


'   Start a command
    CadInputQueue.SendCommand "PLACE REVCLOUD POINTS"

End Sub




Private Sub cmbExistingLevel_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Existing"""

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 0, "MGDSHOOK"

    CadInputQueue.SendCommand "ACTIVE STYLE 0"

    CadInputQueue.SendCommand "ACTIVE WEIGHT 1"

    CommandState.StartDefaultCommand
End Sub





Private Sub cmdEsizeFence_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
```

```
    CadInputQueue.SendCommand "PLACE FENCE ICON"

'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.493826490298015, 30.0246028833885, 0#), 1

'   Coordinates are in master units
    startPoint.X = 0#
    startPoint.Y = 30#
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(41.97131768619, 1.17293206833481E-02, 0#), 1

    point.X = startPoint.X + 42.0000000000001
    point.Y = startPoint.Y - 30#
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X + 1.79290742839691
    point.Y = startPoint.Y + 3.46071529917275
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "PRINT MAXIMIZE"

    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 40.9646372074583
    point.Y = startPoint.Y - 30.3439460111004
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 40.801902248909
    point.Y = startPoint.Y - 20.3922465283965
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub

Private Sub cmdFence18000_Click()
  Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"
```

```
'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.098639241090924, 14.4992497162524, 1.4111111111138), 1

'   Coordinates are in master units
    startPoint.X = 0#
    startPoint.Y = 14.6666666666666
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(22.6630469909838, -4.14684863753751E-03, 1.41111111111402), 1

    point.X = startPoint.X + 22.6666666666667
    point.Y = startPoint.Y - 14.6666666666666
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X + 23.1448617967697
    point.Y = startPoint.Y - 6.5622857142857
    point.Z = startPoint.Z + 1.4111111111123
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "FIT VIEW EXTENDED 5"

    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 21.7245776434224
    point.Y = startPoint.Y - 8.34410349586891
    point.Z = startPoint.Z + 1.41111111111421
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 21.6370483654411
    point.Y = startPoint.Y - 14.7128887948723
    point.Z = startPoint.Z + 1.41111111111426
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand

End Sub




Private Sub cmdNewOrRevLevel_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
```

```
    CadInputQueue.SendCommand "ACTIVE LEVEL ""New or Revisions"""

    CadInputQueue.SendCommand "ACTIVE STYLE 0"

    CadInputQueue.SendCommand "ACTIVE WEIGHT 1"

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 7, "MGDSHOOK"

    CommandState.StartDefaultCommand
End Sub

Private Sub cmdSetRadius_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long
    Dim TxtHeight As Long

    'ActiveSettings.TextStyle.Height
    TxtHeight = ActiveSettings.TextStyle.Height

'   Start a command
    CadInputQueue.SendCommand "PLACE REVCLOUD POINTS"

'   Set a variable associated with a dialog box
    SetCExpressionValue "cloudParams.flags.lockRadius", 1, "COMPCURV"

    'SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * 0.1), "COMPCURV"

    SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * ActiveSettings.TextStyle.Height), "COMPCURV"

    CommandState.StartDefaultCommand

End Sub


Private Sub cmdTextDialog_Click()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro1ModalHandler5
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Preferences [descartes]"

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD USERPREF"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD SPELLCHECK"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
```

```
Private Sub cmdTitleBlock_Click()
  Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command

    CadInputQueue.SendCommand "ACTIVE LEVEL ""Border-titleblock"""

'   Set a variable associated with a dialog box
   ' SetCExpressionValue "tcb->symbology.color", -1, "MGDSHOOK"

    'CadInputQueue.SendCommand "ACTIVE WEIGHT 0"

    CommandState.StartDefaultCommand
End Sub

Private Sub cmdWordProcessor_Click()
  Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro2ModalHandler1
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Preferences [descartes]"

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD USERPREF"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD SPELLCHECK"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub


Private Sub CommandButton1_Click()
 Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "ACTIVE LEVEL ""Notes and References"""

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->symbology.color", 7, "MGDSHOOK"

    CommandState.StartDefaultCommand
End Sub
```

F3 - 7

```
Private Sub UserForm_Click()

End Sub
```

```
Option Base 1
Option Explicit
Option Compare Text   'This fixes issue with MIXED CASE levels --> Dimensions & DIMENSIONS

Sub test_FoundAllLevels()
  Dim rMsg As String
  Dim Critical As Boolean
  rMsg = CheckLevels()

    If rMsg <> "" Then
       MsgBox rMsg  'LEVELS PROBLEM
      Else
         'do nothing2121
    End If

End Sub
Function CheckLevels() As String  'return True if all levels found

  Dim myLevel As Level
  Dim LevelCounter As Integer
  Dim ICountLevels As Integer
  Dim Icount As Integer
  Dim ICountChange As Boolean


  Dim strLevel(23) As String
  'LEVELS ARE CASE SENSITIVE
  strLevel(1) = "Text"
  strLevel(2) = "Property line"
  strLevel(3) = "Backcircle"
  strLevel(4) = "Border-titleblock"
  strLevel(5) = "Dimensions"
  strLevel(6) = "New or Revisions"
  strLevel(7) = "Baselines"
  strLevel(8) = "Fence"
  strLevel(9) = "Removal or Abandoned"
  strLevel(10) = "Contours 1 ft"
  strLevel(11) = "Contours 5 ft"
  strLevel(12) = "Liner Seal to Concrete"
  strLevel(13) = "Liner Extent"
  strLevel(14) = "Notes and References"
  strLevel(15) = "Material Item"
  strLevel(16) = "Vendor"
  strLevel(17) = "Design Master(Red)"
  strLevel(18) = "Existing"
  strLevel(19) = "Mark List"
  strLevel(20) = "Default"
  strLevel(21) = "Fence Corners"
  strLevel(22) = "Centerlines"
  strLevel(23) = "Foundations"

  Icount = 1
  ICountLevels = 0
  'Go thru all levels
  For Each myLevel In ActiveDesignFile.Levels
        'Go thru all levels in strLevel() array and count any that match
        For Icount = 1 To UBound(strLevel)
          If myLevel.Name = strLevel(Icount) Then
             'Debug.Print myLevel.Name
             ICountLevels = ICountLevels + 1
             Exit For
          Else
             'do nothing
          End If
        Next Icount
  Next

  If ICountLevels = UBound(strLevel) Then
     'all substation levels found, but check for excess levels
         If ActiveDesignFile.Levels.count > (UBound(strLevel) + 11) Then
             CheckLevels = "(Excess Levels!) "
         End If
      Else
         CheckLevels = "MISSING LEVELS! "
         If ActiveDesignFile.Levels.count > (UBound(strLevel) + 11) Then
             CheckLevels = " MISSING LEVELS & Excess levels! "
         End If
  End If

End Function
```

```
Option Base 1
Declare Function mdlDialog_fileOpen Lib _
"stdmdlbltin.dll" (ByVal _
 filename As String, ByVal rFileH As Long, ByVal _
 resourceId As Long, ByVal suggestedFileName As String, _
 ByVal filterString As String, _
 ByVal defaultDirectory As String, _
 ByVal titleString As String) As Long

Sub NextFile()
  frmNextFile.show vbModeless
End Sub

Sub NextFileIncrement()
  'Check to see if NextFile form is open

  If frmNextFile.Visible = False Then
    NextFile
    Exit Sub
  End If

  frmNextFile.IncrementDown

End Sub

Sub TestInsertFileLinesToArray()
   Dim fpath As String
   fpath = GetFileListPath
   InsertFileLinesToArray (fpath)
End Sub

Sub InsertFileLinesToArray(filepath As String)
  'this still needs to be updated
  'the routine is to be used to load an array and then to load a list box
  'will use a two-dimensional array so that a number is associated with each
  'file name, so that the file can be opened and the current file can be saved to
  'a text file ... this has yet to be coded.

  Dim myArray() As String    ' Declare dynamic array.
  Dim FileToOpen As String
  Dim I As Integer
  Dim arraySize As Integer
  Dim X As Integer
  Dim a As Integer
  Dim batchfile As String

  'clear ListBox1 before adding items to it
  frmNextFile.ListBox1.Clear

  Dim FFile As Long
  FFile = FreeFile
  'BatchFile = "C:\filelist.txt"
  batchfile = filepath
  Open batchfile For Input As #FFile
  I = 1
  While EOF(FFile) = False
     Line Input #FFile, FileToOpen
     'MsgBox FileToOpen
     'insert the files to an array
        ReDim Preserve myArray(I)      ' Re-allocate
        myArray(I) = FileToOpen    ' Initialize array.
        'MsgBox "array has:" & myArray(i)
        'frmNextFile.ListBox1.AddItem (myArray(i))
        I = I + 1
  Wend

     'loads all files
     frmNextFile.ListBox1.List() = myArray


    ' x = UBound(myArray)
     'a = LBound(myArray)
     'MsgBox Str(x) & " " & Str(a)
     'close the file
     Close FFile
End Sub

Function OpenFile(filename As String, EditMode As Boolean) As Boolean
  Application.OpenDesignFile filename, EditMode
End Function


Sub TestPickAFolder()
  Dim Folder As String
  Dim path As String
  Folder = PickAFolder
  MsgBox RootFolder
End Sub

Sub TestDesktopPathFunction()
   Dim strLine As String
   Dim strPath As String
   strPath = DesktopPath
   MsgBox strPath
End Sub


Function DesktopPath() As String
   Dim objFolders As Object
   Set objFolders = CreateObject("wScript.Shell").specialfolders
   DesktopPath = objFolders("desktop")
End Function

Function fileNamesInTextFile()
    Dim filepath As String
    Dim filename As String
    Dim First As Boolean
    Dim count As Integer
    Dim Folderpath As String
    Dim myFSO As New Scripting.FileSystemObject
    Dim myFolder As Scripting.Folder
    Dim myFile As Scripting.File
    Dim RootFolder As String
    'RootFolder = InputBox("Enter Root Folder:")
    RootFolder = PickAFolder(Folderpath)

    Set myFolder = myFSO.GetFolder(RootFolder)
    First = True
    count = 1
    For Each myFile In myFolder.Files
       Select Case UCase(Right(myFile.Name, 3))
         Case "DGN"
           If First = True Then
             Open filepath For Output As #1
               Print #1, Str(count) & myFile.path
               First = False
             Close #1
           Else
             Open filepath For Append As #1
               Print #1, Str(count) & " " & Right(myFile.path, 14)
             Close #1
           Else
             Open filepath For Append As #1
               Print #1, Str(count) & " " & Right(myFile.path, 14)  'myFile.Path
             Close #1
           End If
           count = count + 1
    End Select
    Next

End Function

Sub SelectDTopFile()

  Dim fname As String
  Dim strPath As String
  Dim FilesFolder As String
  Dim objFolders As Object
  Set objFolders = CreateObject("wScript.Shell").specialfolders

  DTopPath = objFolders("desktop")

  strPath = DTopPath & "\"      'uses function in this module, string path could be hard coded

  strPath = strPath & "Filelists"
  If (Dir(strPath, vbDirectory) <> "") Then
     'do nothing
    Else
      MkDir (strPath)
  End If
  strPath = strPath & "\"
  'fname = SelectFile(strPath, "*.txt", "filelist.txt", "Select the file names file")
  'fname = SelectFile(strPath, "*.txt", "key-ins.txt")
  'MsgBox strPath
End Sub

Sub test_SelectDGNFolder()
  Dim t As String
  t = SelectDGNFolder
  t = t & "did it show up"
  MsgBox t
End Sub
Function SelectDGNFolder() As String
  'This function calls the SelectFile function listed below

  Dim strPath As String

  Dim CurrentFile As DesignFile
  Set CurrentFile = Application.ActiveDesignFile

  strPath = CurrentFile.path & "\"

  SelectDGNFolder = strPath

End Function



Function SelectDGNFolder_archived() As String
  'This function calls the SelectFile function listed below

  Dim fname As String
  Dim strPath As String
  Dim PPath As String
  Dim TitleInfo As String
  Dim AFolderName As String
  Dim BSlash As Integer

  TitleInfo = "To Select a Folder, Select a File inside of a Folder!"

  PPath = "P:\Active Projects\PGE\Substation\"

  strPath = PPath      'uses function in this module, string path could be hard coded

  fname = SelectFile(strPath, "*.dgn", "To select a folder, select a file inside a folder", TitleInfo)
  'MsgBox fname

  If fname = "" Then
    MsgBox "you did not select a file!"
    'clear the list box
    frmNextFile.ListBox1.Clear
    Exit Function
  End If

  BSlash = RightMostBackSlash(fname)
  AFolderName = FolderName(BSlash, fname)
  SelectDGNFolder = AFolderName

End Function
Function SelectFile_archived(strStartingPath As String, strFilter As String, strSuggFName As String, TitleText As String) As String
  'this subroutine requires the declaration statement at the top of this module
  'This routine uses the function SelectDGNFolder listed above
  Dim strFName As String
  Dim lngfhandle As Long
  Dim lngrid As Long
  Dim retVal As Long
  Dim strPath As String
  strFName = Space(255)
  retVal = mdlDialog_fileOpen(filename:=strFName, rFileH:=lngfhandle, resourceId:=lngrid, _
                              suggestedFileName:=strSuggFName, filterString:=strFilter, defaultDirectory:=strStartingPath, _
                              titleString:=TitleText)

  Select Case retVal
     Case 0  'Open
       strFName = Left(strFName, InStr(1, strFName, Chr(0)) - 0.1)
       'MsgBox "File Selected:" & vbCr & strFName
     Case 1  'Cancel
       MsgBox "No File Selected."
       strFName = ""
     End Select
  SelectFile = strFName
End Function
Function RightMostBackSlash(strPath As String) As Integer
  'This function is used by the SelectDGNFolder listed above
  Dim count As Integer
  Dim LeftPart As String
  Dim RightPart As String
  count = 0

  'find the right most "\" backslash
  While LeftPart <> "\"
    count = count + 1
    RightPart = Right(strPath, count)
    'Debug.Print RightPart
    LeftPart = Left(RightPart, 1)
    'Debug.Print LeftPart
  Wend
  'MsgBox Str(count)
  RightMostBackSlash = count

End Function
Function FolderName(BackSlashPos As Integer, PathAndFileName As String) As String
  'I don't think this function will be used
  Dim Folderpath As String
  Dim filename As String
  Dim intBSlashLoc As Integer

  Trim (PathAndFileName)
  slen = Len(PathAndFileName)

  Folderpath = Left(PathAndFileName, slen - BackSlashPos + 1)
  'FileName = Right(PathAndFileName, intBSlashLoc - 1)
  'MsgBox "Folder Path:" & " " & folderPath & vbLf & "File Name:" & " " & FileName

  FolderName = Folderpath
End Function

Sub test_FileCreate()
   Call FileCreate("C:\Users\knowles_keith\Desktop\Microstation_test_Folder\")
End Sub

Function GetFileListPath() As String
   Dim DesktopPath As String
   Dim objFolders As Object

   Set objFolders = CreateObject("wScript.Shell").specialfolders
   DesktopPath = objFolders("desktop")
   GetFileListPath = DesktopPath & "\Filelists\filelist.txt"

End Function

Sub FileCreate(Folder As String)
  Dim textfile As String
  Dim Folderpath As String
  Dim DesktopPath As String
  Dim objFolders As Object

   Set objFolders = CreateObject("wScript.Shell").specialfolders
   DesktopPath = objFolders("desktop")

   textfile = GetFileListPath

   'folderPath = "C:\Users\knowles_keith\Desktop\Microstation_test_Folder\"
   'Folderpath = "C:\Users\knowles_keith\Desktop\Microstation_test_Folder\"

   Folderpath = Folder

   'folderPath = "P:\Active Projects\PGE\Substation\6446 BELL\2000 Substation\2300 Engineering\2310 Electrical\2311 Drawings\Indoor\
"
   Call TextFileCreate(textfile, Folderpath)

End Sub

Sub TextFileCreate(textfile As String, Folderpath As String)
'*************************************************************
'Good - code is used in button, "Change Folder Path"

  Dim N As Integer
  Dim I As Integer
  Dim MyPath As String
  Dim MyName As String
  Dim FileNames() As String
  Dim counter As Integer

  MyPath = Folderpath

  'Display the names in the directory
    MyName = Dir(MyPath)    'Retrieve the first entry.
    I = 0
    Do While MyName <> ""    ' Start the loop.
        ' Ignore the current directory and the encompassing directory.
        If MyName <> "." And MyName <> ".." Then
          If Right(MyName, 4) = ".dgn" Then
            'Debug.Print MyName    ' Display entry only if it
             I = I + 1
             ReDim Preserve FileNames(I)
             FileNames(I) = MyName
             Debug.Print FileNames(I)
          End If
        End If    ' it represents a directory.
        MyName = Dir    ' Get next entry.
    Loop

    'create textfile or over-write the existing file
    'textfile

    Open textfile For Output As #1  'over-writes and/or creates new file
    Print #1, FileNames(1)
    Close #1

    If UBound(FileNames) <= 1 Then
        Exit Sub
    End If
    counter = 2
      Open textfile For Append As #1
    For counter = 2 To UBound(FileNames)
      Print #1, FileNames(counter)
    Next
    Close #1

 '*****************************************************************************
End Sub

Sub filepathsTxtFile()  'needs to take in the file path name, may need to
  Dim textfile As String
  Dim N As Integer
  Dim I As Integer
  Dim MyPath As String
  Dim MyName As String
  Dim FolderString() As String

  'This folder should be passed to the routine by arguments
  Folderpath = "C:\Users\knowles_keith\Desktop\Microstation_test_Folder\"

  'get file --- usually located on the desktop folder with the name filelist.txt
  'textfile = "c:\filelist.txt"
  MyPath = Folderpath

  'Display the names in the directory
    MyName = Dir(MyPath)    ' Retrieve the first entry
    Do While MyName <> ""    ' Start the loop
        ' Ignore the current directory and the encompassing directory
        If MyName <> "." And MyName <> ".." Then
          If Right(MyName, 4) = ".dgn" Then
            Debug.Print MyName    ' Display entry only if it
          End If
        End If    ' it represents a directory.
        MyName = Dir    ' Get next entry.
    Loop
End Sub

Sub test_SplitFolderFromFileName()

  Dim Folderpath As String
  Dim filename As String
  Dim strPath As String
  Dim intBSlashLoc As Integer

  strPath = "C:\Indoor\123.dgn"

  Trim (strPath)
  slen = Len(strPath)

  'Get the position of the right most back slash in file path
  intBSlashLoc = RightMostBackSlash(strPath)

  Folderpath = Left(strPath, slen - intBSlashLoc + 1)
  filename = Right(strPath, intBSlashLoc - 1)
  MsgBox "Folder Path:" & " " & Folderpath & vbLf & "File Name:" & " " & filename

End Sub

Function WritePathToFile(FullFolderPath As String) As Boolean
   Dim textfilepath As String
   Dim path As String
   Dim IsFolderThere As Boolean

   'get the desktop folder path
   path = DesktopPath
   path = path & "\Filelists"

   'check to see if folder on desktop
   If (Dir(path, vbDirectory) <> "") Then
        IsFolderThere = True
        'MsgBox "folder there"
     Else
        IsFolderThere = False
        MsgBox "Folder not there"
        MkDir (path)
   End If

   path = path & "\path.txt"

   Open path For Output As #1
      Print #1, FullFolderPath
   Close #1

End Function

Function GetPathToFile() As String
   Dim textfilepath As String
   Dim path As String
   Dim IsFolderThere As Boolean

   'get the desktop folder path
   path = DesktopPath
   path = path & "\Filelists"

   'check to see if folder on desktop
   If (Dir(path, vbDirectory) <> "") Then
        IsFolderThere = True
        'MsgBox "folder there"
     Else
        IsFolderThere = False
        'MsgBox "Folder not there"
        MkDir (path)
   End If

   path = path & "\path.txt"

   Open path For Input As #1
      Line Input #1, FullFolderPath
   Close #1

   GetPathToFile = FullFolderPath
End Function
```

```
Public Type RevInfo

   Rev1 As String
   Rev2 As String
   Rev3 As String
   Date_YR As String
   Date_MN As String
   Date_DY As String
   DateA_YR As String
   DateA_MN As String
   DateA_DY As String
   DateB_YR As String
   DateB_MN As String
   DateB_DY As String
   DateC_YR As String
   DateC_MN As String
   DateC_DY As String
   Line1a As String
   line1b As String
   Line2a As String
   Line2b As String
   Line3a As String
   Line3b As String
   Line4a As String
   Line4b As String
   Reg_DRA As String
   Reg_DES As String
   Reg_ENG As String
   Reg_CHK As String
   Reg_APP As String
   A_DRA As String
   A_DES As String
   A_ENG As String
   A_CHK As String
   A_APP As String
   B_DRA As String
   B_DES As String
   B_ENG As String
   B_CHK As String
   B_APP As String
   C_DRA As String
   C_DES As String
   C_ENG As String
   C_CHK As String
   C_APP As String

End Type

Sub RevBlockEditor()
  frmRevBlock.show vbModeless
End Sub
Function GetRevInfo() As RevInfo

    Dim GetTitleInfo As TitleInfo
    Dim newvalue As String
    Dim myTag As TagElement
    Dim myElemEnum As ElementEnumerator
    Dim myFilter As New ElementScanCriteria
    myFilter.ExcludeAllTypes
    myFilter.IncludeType msdElementTypeTag
    Set myElemEnum = Application.ActiveModelReference.Scan(myFilter)

    While myElemEnum.MoveNext
        Set myTag = myElemEnum.Current

        With myTag

            If .TagSetName = "RevNo" And .TagDefinitionName = "Rev1" Then
                GetRevInfo.Rev1 = .Value
            End If

            If .TagSetName = "RevNo" And .TagDefinitionName = "Rev2" Then
                GetRevInfo.Rev2 = .Value
            End If

            If .TagSetName = "RevNo" And .TagDefinitionName = "Rev3" Then
                GetRevInfo.Rev3 = .Value
            End If

            If .TagSetName = "Date" And .TagDefinitionName = "1_Month" Then
                GetRevInfo.Date_MN = .Value
            End If

            If .TagSetName = "Date" And .TagDefinitionName = "2_Day" Then
                GetRevInfo.Date_DY = .Value
            End If

            If .TagSetName = "Date" And .TagDefinitionName = "3_Year" Then
                GetRevInfo.Date_YR = .Value
            End If

            If .TagSetName = "DateA" And .TagDefinitionName = "1_Month" Then
                GetRevInfo.DateA_MN = .Value
            End If

            If .TagSetName = "DateA" And .TagDefinitionName = "2_Day" Then
                GetRevInfo.DateA_DY = .Value
            End If

            If .TagSetName = "DateA" And .TagDefinitionName = "3_Year" Then
                GetRevInfo.DateA_YR = .Value
            End If

            If .TagSetName = "DateB" And .TagDefinitionName = "1_Month" Then
                GetRevInfo.DateB_MN = .Value
            End If

            If .TagSetName = "DateB" And .TagDefinitionName = "2_Day" Then
```

```
                GetRevInfo.DateB_DY = .Value
            End If

            If .TagSetName = "DateB" And .TagDefinitionName = "3_Year" Then
                GetRevInfo.DateB_YR = .Value
            End If

            If .TagSetName = "DateC" And .TagDefinitionName = "1_Month" Then
                GetRevInfo.DateC_MN = .Value
            End If

            If .TagSetName = "DateC" And .TagDefinitionName = "2_Day" Then
                GetRevInfo.DateC_DY = .Value
            End If

            If .TagSetName = "DateC" And .TagDefinitionName = "3_Year" Then
                GetRevInfo.DateC_YR = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line1a" Then
                GetRevInfo.Line1a = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line1b" Then
                GetRevInfo.line1b = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line2a" Then
                GetRevInfo.Line2a = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line2b" Then
                GetRevInfo.Line2b = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line3a" Then
                GetRevInfo.Line3a = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line3b" Then
                GetRevInfo.Line3b = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line4a" Then
                GetRevInfo.Line4a = .Value
            End If

            If .TagSetName = "RevisionDescriptions" And .TagDefinitionName = "Line4b" Then
                GetRevInfo.Line4b = .Value
            End If

            If .TagSetName = "Initals reg" And .TagDefinitionName = "1_Drawn" Then
                GetRevInfo.Reg_DRA = .Value
            End If
            If .TagSetName = "Initals reg" And .TagDefinitionName = "2_Designer" Then
                GetRevInfo.Reg_DES = .Value
            End If

            If .TagSetName = "Initals reg" And .TagDefinitionName = "3_Engineer" Then
                GetRevInfo.Reg_ENG = .Value
            End If

            If .TagSetName = "Initals reg" And .TagDefinitionName = "4_Check" Then
                GetRevInfo.Reg_CHK = .Value
            End If

            If .TagSetName = "Initals reg" And .TagDefinitionName = "5_Approved" Then
                GetRevInfo.Reg_APP = .Value
            End If

            If .TagSetName = "InitialsA_1" And .TagDefinitionName = "1_Drawn" Then
                GetRevInfo.A_DRA = .Value
            End If

            If .TagSetName = "InitialsA_1" And .TagDefinitionName = "2_Designer" Then
                GetRevInfo.A_DES = .Value
            End If

            If .TagSetName = "InitialsA_1" And .TagDefinitionName = "3_Engineer" Then
                GetRevInfo.A_ENG = .Value
            End If

            If .TagSetName = "InitialsA_1" And .TagDefinitionName = "4_Check" Then
                GetRevInfo.A_CHK = .Value
            End If

            If .TagSetName = "InitialsA_1" And .TagDefinitionName = "5_Approved" Then
                GetRevInfo.A_APP = .Value
            End If


            If .TagSetName = "InitialsB_1" And .TagDefinitionName = "1_Drawn" Then
                GetRevInfo.B_DRA = .Value
            End If

            If .TagSetName = "InitialsB_1" And .TagDefinitionName = "2_Designer" Then
                GetRevInfo.B_DES = .Value
            End If

            If .TagSetName = "InitialsB_1" And .TagDefinitionName = "3_Engineer" Then
                GetRevInfo.B_ENG = .Value
            End If

            If .TagSetName = "InitialsB_1" And .TagDefinitionName = "4_Check" Then
                GetRevInfo.B_CHK = .Value
            End If
            If .TagSetName = "InitialsB_1" And .TagDefinitionName = "5_Approved" Then
                GetRevInfo.B_APP = .Value
            End If


            If .TagSetName = "InitialsC_1" And .TagDefinitionName = "1_Drawn" Then
                GetRevInfo.C_DRA = .Value
            End If

            If .TagSetName = "InitialsC_1" And .TagDefinitionName = "2_Designer" Then
                GetRevInfo.C_DES = .Value
            End If

            If .TagSetName = "InitialsC_1" And .TagDefinitionName = "3_Engineer" Then
                GetRevInfo.C_ENG = .Value
            End If

            If .TagSetName = "InitialsC_1" And .TagDefinitionName = "4_Check" Then
                GetRevInfo.C_CHK = .Value
            End If

            If .TagSetName = "InitialsC_1" And .TagDefinitionName = "5_Approved" Then
                GetRevInfo.C_APP = .Value
            End If

        End With

     Wend

     GetRevInfo = GetRevInfo

End Function

Sub UpdateRevInfo(TagSetName As String, TagDefName As String, TagValue As String)

    Dim newvalue As String
    Dim myTag As TagElement
    Dim myElemEnum As ElementEnumerator
    Dim myFilter As New ElementScanCriteria
    myFilter.ExcludeAllTypes
    myFilter.IncludeType msdElementTypeTag
    Set myElemEnum = Application.ActiveModelReference.Scan(myFilter)

    While myElemEnum.MoveNext
        Set myTag = myElemEnum.Current

        If myTag.TagSetName = TagSetName And myTag.TagDefinitionName = TagDefName Then
            myTag.Value = TagValue
            myTag.Rewrite
        End If
     Wend
End Sub
```

```
Public Type TitleInfo
   SubName As String
   IndexNo As String
   SheetNum As String
   SheetOf As String
   RevNum As String
   DistCode As String
   DescLine1 As String
   DescLine2 As String
   DescLine3 As String
End Type

Function GetTagInfo() As TitleInfo

    Dim GetTitleInfo As TitleInfo
    Dim newvalue As String
    Dim myTag As TagElement
    Dim myElemEnum As ElementEnumerator
    Dim myFilter As New ElementScanCriteria
    myFilter.ExcludeAllTypes
    myFilter.IncludeType msdElementTypeTag
    Set myElemEnum = Application.ActiveModelReference.Scan(myFilter)

    While myElemEnum.MoveNext
        Set myTag = myElemEnum.Current

        With myTag

            If .TagSetName = "SubstationName" And .TagDefinitionName = "SubstationName" Then
                GetTitleInfo.SubName = .Value
            End If

            If .TagSetName = "SubstationIndexNo" And .TagDefinitionName = "DocumentNo" Then
                GetTitleInfo.IndexNo = .Value
            End If

            If .TagSetName = "SheetNo" And .TagDefinitionName = "Number" Then
                GetTitleInfo.SheetNum = .Value
            End If

            If .TagSetName = "SheetOf" And .TagDefinitionName = "of #" Then
                GetTitleInfo.SheetOf = .Value
            End If

            If myTag.TagSetName = "RevisionNo" And .TagDefinitionName = "Number" Then
                GetTitleInfo.RevNum = .Value
            End If

            If .TagSetName = "DistributionCode" And .TagDefinitionName = "Code" Then
                GetTitleInfo.DistCode = .Value
            End If

            If .TagSetName = "DescriptionLines" And .TagDefinitionName = "Line1" Then
                GetTitleInfo.DescLine1 = .Value
            End If

            If .TagSetName = "DescriptionLines" And .TagDefinitionName = "Line2" Then
                GetTitleInfo.DescLine2 = .Value
            End If

            If .TagSetName = "DescriptionLines" And .TagDefinitionName = "Line3" Then
                GetTitleInfo.DescLine3 = .Value
            End If


        End With

     Wend

     GetTagInfo = GetTitleInfo


End Function

Sub TitleBlock()
  frmTitleBlock.show vbModeless
End Sub

Sub TEST_UpdateTagInfo()

 UpdateTagInfo "SubstationIndexNo", "DocumentNo", "RVGT-1234"
 UpdateTagInfo "SubstationName", "SubstationName", "hello everybody"

End Sub

Sub UpdateTagInfo(TagSetName As String, TagDefName As String, TagValue As String)

    Dim newvalue As String
    Dim myTag As TagElement
    Dim myElemEnum As ElementEnumerator
    Dim myFilter As New ElementScanCriteria
    myFilter.ExcludeAllTypes
    myFilter.IncludeType msdElementTypeTag
    Set myElemEnum = Application.ActiveModelReference.Scan(myFilter)

    While myElemEnum.MoveNext
        Set myTag = myElemEnum.Current

        If myTag.TagSetName = TagSetName And myTag.TagDefinitionName = TagDefName Then
            'myTag.Value = TagUpdate.SubName
            myTag.Value = TagValue
            myTag.Rewrite
        End If
     Wend
End Sub


Sub GetTagsA()
```

```
'used to get all of the tags out of a drawing
    Dim newvalue As String
    Dim myTag As TagElement
    Dim myElemEnum As ElementEnumerator
    Dim myFilter As New ElementScanCriteria
    myFilter.ExcludeAllTypes
    myFilter.IncludeType msdElementTypeTag
    Set myElemEnum = Application.ActiveModelReference.Scan(myFilter)

    While myElemEnum.MoveNext
        Set myTag = myElemEnum.Current
        'MsgBox myTag.TagSetName & vbTab & myTag.TagDefinitionName
        Debug.Print myTag.TagSetName & vbTab & vbTab & myTag.TagDefinitionName _
            & vbTab & myTag.Value
    Wend
            'newvalue = InputBox("enter new value for: " & myTag.TagDefinitionName)
            'myTag.Value = newvalue
            'myTag.Rewrite

End Sub
```

```
Sub Macro1()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro1ModalHandler5
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Preferences [descartes]"

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD USERPREF"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD SPELLCHECK"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
Sub Macro2()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro2ModalHandler1
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Preferences [descartes]"

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD USERPREF"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD SPELLCHECK"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
Sub Macro3()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "DIALOG PLOT"

    Dim modalHandler As New Macro3ModalHandler
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Print - Raster Options"

    CadInputQueue.SendCommand "PRINT ROPTSDIALOG"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
```

```
Sub Macro4()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Send a keyin that can be a command string
    CadInputQueue.SendKeyin "level purge all"

    Dim modalHandler As New Macro4ModalHandler1
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Design File Settings"

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"

    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
Sub Macro5()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro5ModalHandler0
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Color Table"

'   Start a command
    CadInputQueue.SendCommand "DIALOG COLOR"

'   Coordinates are in master units
    startPoint.X = 2.95957877203563
    startPoint.Y = 0.120543355820554
    startPoint.Z = 0.083333333333315

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.178317424247461
    point.Y = startPoint.Y + 0.414411330316334
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "DELETE ELEMENT"
    point.X = startPoint.X - 0.111822281528084
    point.Y = startPoint.Y - 5.83859518712345E-02
    point.Z = startPoint.Z - 2.3592E-16
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.241407279381207
    point.Y = startPoint.Y + 0.130217249126066
    point.Z = startPoint.Z - 2.3592E-16
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "DELETE ELEMENT"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD VBAPM"

'   The following statement opens modal dialog "Color Table"

    CadInputQueue.SendCommand "DIALOG COLOR"

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"

    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

'   The following statement opens modal dialog "Level/Filter Import"

'   The following statement opens modal dialog "Import Levels"

    CadInputQueue.SendCommand "LEVELMANAGER LIBRARY IMPORT"

    point.X = startPoint.X + 0.40227260475232
    point.Y = startPoint.Y - 0.312869042962989
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.713249564377427
    point.Y = startPoint.Y - 2.95800476295584E-02
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

'   Set a variable associated with a dialog box
    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 2.55371618863213
    point.Y = startPoint.Y + 1.78409905360514
    point.Z = startPoint.Z + 1.80411E-15
    CadInputQueue.SendDataPoint point, 1
    point.X = startPoint.X + 0.213079974691378
    point.Y = startPoint.Y - 0.414583282239851
    point.Z = startPoint.Z + 1.80411E-15
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 3.5085772346024
    point.Y = startPoint.Y + 2.19851038392148
    point.Z = startPoint.Z + 1.80411E-15
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.207327799715654
    point.Y = startPoint.Y - 0.35702615302925
    point.Z = startPoint.Z + 1.80411E-15
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "Change Attributes"

'   Send a keyin that can be a command string
    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES USEACTIVE ON"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE LEVEL"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LEVEL ""New or Revisions"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE COLOR"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET COLOR ""0"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE LINESTYLE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LINESTYLE ""Continuous"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE WEIGHT"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET WEIGHT 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TRANSPARENCY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TRANSPARENCY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE PRIORITY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET PRIORITY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE ELEMENTCLASS"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET ELEMENTCLASS PRIMARY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TEMPLATE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TEMPLATE """""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES MAKECOPY OFF"
    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENTIREELEMENT OFF"

    SetCExpressionValue "tcb->msToolSettings.general.useFence", 0, "CHANGEATTRIBS"

    CadInputQueue.SendCommand "LOCK FENCE INSIDE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LEVEL ""Vendor"""

    point.X = startPoint.X + 0.115293000104061
    point.Y = startPoint.Y + 0.926497828367175
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 3.20191440620909
    point.Y = startPoint.Y + 1.94525901539483
    point.Z = startPoint.Z + 2.34535E-15
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.631411213932183
    point.Y = startPoint.Y - 6.28739690214429E-03
    point.Z = startPoint.Z + 2.34535E-15
    CadInputQueue.SendDataPoint point, 1

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendCommand "NEWFILE U:\New folder\bellSWGRTITLEBLOCK.dgn"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    point.X = startPoint.X - -39.6470393651277
    point.Y = startPoint.Y - 5.9953998274684
    point.Z = startPoint.Z + 2.45666666666633
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -17.8879261258157
    point.Y = startPoint.Y + 13.083591904566
    point.Z = startPoint.Z + 2.45666666666633
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MDL LOAD CLIPBRD COPY"
    CadInputQueue.SendCommand "NEWFILE ""P:\Active Projects\PGE\Substation\6446 BELL\2000 Substation\2300 Engineering\2310 Electrica
l\2311 Drawings\Indoor\bell7313a0.dgn"",""~4683"""

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "MDL KEYIN CLIPBRD CLIPBOARD PASTE"

    point.X = startPoint.X - 28.631953394003
    point.Y = startPoint.Y + 16.2454221816907
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

'   Send a reset to the current command
    CadInputQueue.SendReset

    point.X = startPoint.X - -9.72067325664033
    point.Y = startPoint.Y - 10.5493999921612
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MDL SILENTLOAD USTNVBA MACROS"

    CadInputQueue.SendReset

    point.X = startPoint.X - 6.70546566878871
    point.Y = startPoint.Y + 56.2127948024225
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 93.0284853312103
    point.Y = startPoint.Y - 26.450259538265
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "SCALE ICON"

    CadInputQueue.SendCommand "ACTIVE XSCALE 0.3900"

    CadInputQueue.SendCommand "ACTIVE SCALE"

    point.X = startPoint.X - -15.9807978633999
    point.Y = startPoint.Y - 9.31802029667172
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset
    CadInputQueue.SendCommand "MOVE ICON"

'   Send a tentative point
    CadInputQueue.SendTentativePoint Point3dFromXYZ(43.3222605049795, 15.6762988369673, 2.53999999999963), 1

    point.X = startPoint.X - -40.2932509755801
    point.Y = startPoint.Y + 15.5668957102602
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-10.8485751025206, 26.7805644149428, 0#), 1

    point.X = startPoint.X - 13.757003394003
    point.Y = startPoint.Y + 26.7454221816907
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 17.8581020925771
    point.Y = startPoint.Y + 6.35711818150786
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 18.0026820661479
    point.Y = startPoint.Y + 6.73084617122634
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 2, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR MODE REMOVE"

    point.X = startPoint.X - 19.2263406966295
    point.Y = startPoint.Y + 5.90013022633797
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 19.4851915607698
    point.Y = startPoint.Y + 7.26581970489595
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MOVE ICON"

    point.X = startPoint.X - 18.457003394003
    point.Y = startPoint.Y + 6.8954221816907
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 17.7338536777898
    point.Y = startPoint.Y + 7.65405225873153
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "MOVE ICON"

    point.X = startPoint.X - 17.4179055416471
    point.Y = startPoint.Y + 7.88136182113021
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 17.0881472557046
    point.Y = startPoint.Y + 6.44033105421862
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 18.0520137461763
    point.Y = startPoint.Y + 5.93398990685809
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 22.0994999854615
    point.Y = startPoint.Y + 11.4532591098579
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 1, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR MODE ADD"

    point.X = startPoint.X - 17.9014096070401
    point.Y = startPoint.Y + 5.91515281060212
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 22.1936275724216
    point.Y = startPoint.Y + 11.2837252435542
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MOVE ICON"
    point.X = startPoint.X - 21.8924192941492
    point.Y = startPoint.Y + 11.1330284735064
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 21.9213352888634
    point.Y = startPoint.Y + 11.8660175630188
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendCommand "EDIT SINGLE DIALOG"

    point.X = startPoint.X - 12.979263944635
    point.Y = startPoint.Y + 8.66307073823081
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

'   Send a message string to an application
'   Content is defined by the application
    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine BELL SUBSTATION #3                                      "

    point.X = startPoint.X - 12.979263944635
    point.Y = startPoint.Y + 8.66307073823081
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - 12.8346839710643
    point.Y = startPoint.Y + 8.8197953790805
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine 15KV OUTDOOR SWITCHGEAR                                      "

    point.X = startPoint.X - 12.8105873088025
    point.Y = startPoint.Y + 8.8197953790805
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - 12.4491373748756
    point.Y = startPoint.Y + 8.85596260389197
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine UNIT 4 - FDR. BKR R304 CONTROL SCHEM.
         "

    point.X = startPoint.X - 12.4491373748756
    point.Y = startPoint.Y + 8.85596260389197
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - 11.1479176127389
    point.Y = startPoint.Y + 8.94035279511872
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 12.7764830014951
    point.Y = startPoint.Y + 7.54338620193733
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine 3 "

    point.X = startPoint.X - 12.7764830014951
    point.Y = startPoint.Y + 7.54338620193733
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - 13.3702247596256
    point.Y = startPoint.Y + 7.41221973328774
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine BELL-7313"

    point.X = startPoint.X - 14.4343333651063
    point.Y = startPoint.Y + 7.12673977210923
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 5, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR ALL"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"
    CadInputQueue.SendKeyin "VBA RUN BUTTONS"

    point.X = startPoint.X - 57.5043080250155
    point.Y = startPoint.Y + 33.4859425904581
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 8.85056034016884
    point.Y = startPoint.Y - -0.431345583671012
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-44.1353407261628, 5.16875384750476, 2.53999999999963), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-43.043773250429, 4.99629508051283, 2.53999999999963), 1

    point.X = startPoint.X - 45.757003394003
    point.Y = startPoint.Y + 5.2454221816907
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X - 48.0141342146059
    point.Y = startPoint.Y + 7.92252327488308
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendKeyin Chr$(27)

    CadInputQueue.SendCommand "MOVE ICON"

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-42.9458768831316, 4.95628464657069, 2.53999999999963), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-42.7105579157313, 5.42721205296999, 2.53999999999963), 1

    point.X = startPoint.X - 45.757003394003
    point.Y = startPoint.Y + 5.2454221816907
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendKeyin "xy=0,0"

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "VIEW ON 5"

    CadInputQueue.SendKeyin "dialog viewsettings popup"

    CadInputQueue.SendKeyin "MDL KEYIN BENTLEY.VIEWATTRIBUTESDIALOG,VAD VIEWATTRIBUTESDIALOG SETATTRIBUTE 0 DataFields False"

    point.X = startPoint.X - -31.2800461097629
    point.Y = startPoint.Y + 4.42010573726455
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1
    point.X = startPoint.X - -32.3080604127794
    point.Y = startPoint.Y + 6.25881152715817
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "PRINT EXECUTE"

    point.X = startPoint.X - -32.1410080885392
    point.Y = startPoint.Y + 6.49025701120072
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - -31.7426525461203
    point.Y = startPoint.Y + 6.2973857744986
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

'   The following statement opens modal dialog "Open"

    CadInputQueue.SendCommand "DIALOG OPENFILE"

    point.X = startPoint.X - -28.1960032007134
    point.Y = startPoint.Y + 4.6386931388603
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - -27.3221910431493
    point.Y = startPoint.Y - -2.54282570003051
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

'   The following statement opens modal dialog "Open"

    CadInputQueue.SendCommand "DIALOG OPENFILE"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"
    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    point.X = startPoint.X + 0.193136870243662
    point.Y = startPoint.Y - 0.277476094811245
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.285178612605209
    point.Y = startPoint.Y + 0.491060597448877
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "DELETE ELEMENT"

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

'   The following statement opens modal dialog "Color Table"

    CadInputQueue.SendCommand "DIALOG COLOR"

    point.X = startPoint.X - 3.6622037756937
    point.Y = startPoint.Y + 2.70297650225928
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD USTNVBA MACROS"

    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

    CadInputQueue.SendKeyin "level purge all"

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"

    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

'   The following statement opens modal dialog "Alert"

    CadInputQueue.SendCommand "UNDO ALL"

    point.X = startPoint.X + 1.02681143702998
    point.Y = startPoint.Y + 0.608699634550071
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.99088028597463
    point.Y = startPoint.Y + 0.698582761919565
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 1.04477701255766
    point.Y = startPoint.Y + 0.761500951078211
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    point.X = startPoint.X + 0.936983559391602
    point.Y = startPoint.Y + 0.700380424466956
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.784276167406347
    point.Y = startPoint.Y + 0.853181740995096
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.125320921559938
    point.Y = startPoint.Y - 0.256515349508679
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.280543494119067
    point.Y = startPoint.Y + 0.117398460348417
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "DELETE ELEMENT"

'   The following statement opens modal dialog "Color Table"

    CadInputQueue.SendCommand "DIALOG COLOR"

    CadInputQueue.SendKeyin "level purge all"

    point.X = startPoint.X - 2.81984285313975
    point.Y = startPoint.Y + 1.81724569437845
    point.Z = startPoint.Z - 1.388E-17
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.672215890052462
    point.Y = startPoint.Y - 1.3736053272386
    point.Z = startPoint.Z - 1.388E-17
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendKeyin Chr$(27)
    CadInputQueue.SendCommand "SCALE ICON"

    CadInputQueue.SendCommand "ACTIVE XSCALE 0.3900"

    CadInputQueue.SendCommand "ACTIVE SCALE"

    point.X = startPoint.X - 2.00016346968949
    point.Y = startPoint.Y - 0.800600390258071
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"

    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

'   The following statement opens modal dialog "Color Table"

    CadInputQueue.SendCommand "DIALOG COLOR"

    point.X = startPoint.X - 0.385769666542224
    point.Y = startPoint.Y - 0.422631053694063
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.459356663903589
    point.Y = startPoint.Y - 0.290092989400102
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 2.44252624279237
    point.Y = startPoint.Y + 0.265830558055125
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1
    point.X = startPoint.X - 0.830971000578482
    point.Y = startPoint.Y - 0.919648794796418
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

'   The following statement opens modal dialog "Level/Filter Import"

'   The following statement opens modal dialog "Import Levels"

    CadInputQueue.SendCommand "LEVELMANAGER LIBRARY IMPORT"

    point.X = startPoint.X - 0.301144619576654
    point.Y = startPoint.Y - 0.595666859855624
    point.Z = startPoint.Z + 0#
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "Change Attributes"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES USEACTIVE ON"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE LEVEL"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LEVEL ""Vendor"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE COLOR"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET COLOR ""0"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE LINESTYLE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LINESTYLE ""Continuous"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE WEIGHT"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET WEIGHT 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TRANSPARENCY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TRANSPARENCY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE PRIORITY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET PRIORITY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE ELEMENTCLASS"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET ELEMENTCLASS PRIMARY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TEMPLATE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TEMPLATE """""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES MAKECOPY OFF"
    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENTIREELEMENT OFF"

    SetCExpressionValue "tcb->msToolSettings.general.useFence", 0, "CHANGEATTRIBS"

    CadInputQueue.SendCommand "LOCK FENCE INSIDE"

    point.X = startPoint.X - 0.551340410605294
    point.Y = startPoint.Y - 0.367406860238246
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 2.21647618527293
    point.Y = startPoint.Y + 0.190587594471574
    point.Z = startPoint.Z + 7.0777E-16
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 1.18165903487874
    point.Y = startPoint.Y - 0.913896274644771
    point.Z = startPoint.Z + 7.0777E-16
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendCommand "NEWFILE U:\New folder\bellSWGRTITLEBLOCK.dgn"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    point.X = startPoint.X - 6.22655504146303
    point.Y = startPoint.Y + 26.8878035695086
    point.Z = startPoint.Z + 2.45666666666633
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -37.5159921922054
    point.Y = startPoint.Y - 8.57667517944953
    point.Z = startPoint.Z + 2.45666666666633
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MDL LOAD CLIPBRD COPY"

    point.X = startPoint.X - -33.2538978463608
    point.Y = startPoint.Y - 11.6068679839491
    point.Z = startPoint.Z + 2.45666666666633
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -32.6930959587497
    point.Y = startPoint.Y - 10.5968037157826
    point.Z = startPoint.Z + 2.45666666666633
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -25.5148317973271
    point.Y = startPoint.Y - 11.1579505314307
    point.Z = startPoint.Z + 2.45666666666633
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "NEWFILE P:\Active Projects\PGE\Substation\6446 BELL\2000 Substation\2300 Engineering\2310 Electrical\
2311 Drawings\Indoor\bell7313b0.dgn"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "MDL KEYIN CLIPBRD CLIPBOARD PASTE"

    CadInputQueue.SendCommand "ACTIVE ANGLE 0.0000°"

    CadInputQueue.SendCommand "ACTIVE ANGLE"

    CadInputQueue.SendCommand "ACTIVE XSCALE 1.0000"

    CadInputQueue.SendCommand "ACTIVE SCALE"

    point.X = startPoint.X - 4.70169394288028
    point.Y = startPoint.Y - 1.99909755251633
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    point.X = startPoint.X - -15.792365326435
    point.Y = startPoint.Y + 10.7901239732489
    point.Z = startPoint.Z + 2.45666666666614
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -51.9735448359856
    point.Y = startPoint.Y - 19.4560095039993
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MOVE ICON"
    CadInputQueue.SendTentativePoint Point3dFromXYZ(49.5509086694719, 8.81060678696583, 2.53999999999945), 1

    point.X = startPoint.X - -46.5778380614257
    point.Y = startPoint.Y + 8.53319221428959
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(13.1579450193187, 8.54097098153532, 0#), 1

    point.X = startPoint.X - -10.1732560571197
    point.Y = startPoint.Y + 8.50090244748367
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendKeyin "VBA RUN BUTTONS"

    CadInputQueue.SendCommand "VIEW ON 5"

    CadInputQueue.SendCommand "PLACE FENCE ICON"

    point.X = startPoint.X - 25.7595787720356
    point.Y = startPoint.Y + 12.5794566441794
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    point.X = startPoint.X - 29.0959011922633
    point.Y = startPoint.Y + 11.981172036036
    point.Z = startPoint.Z - -2.45666666666612
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -15.9025063967995
    point.Y = startPoint.Y - 22.2194920782503
    point.Z = startPoint.Z - -2.45666666666612
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MOVE ICON"

    point.X = startPoint.X - 21.8267439428803
    point.Y = startPoint.Y - 12.9990975525163
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1
    CadInputQueue.SendKeyin "xy=0,0"

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendCommand "NEWFILE U:\New folder\bellSWGRTITLEBLOCK.dgn"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

'   The following statement opens modal dialog "Compress Options"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

'   The following statement opens modal dialog "Level/Filter Import"

'   The following statement opens modal dialog "Import Levels"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    point.X = startPoint.X - -31.410299380563
    point.Y = startPoint.Y + 4.80849955322854
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -30.6764212172336
    point.Y = startPoint.Y + 5.59931598527109
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -27.1952042886195
    point.Y = startPoint.Y - -1.43811523571386
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -26.7059521797332
    point.Y = startPoint.Y - -2.66199780911304
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1
    CadInputQueue.SendCommand "Change Attributes"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES USEACTIVE ON"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE LEVEL"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LEVEL ""Vendor"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE COLOR"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET COLOR ""0"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE LINESTYLE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LINESTYLE ""Continuous"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE WEIGHT"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET WEIGHT 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TRANSPARENCY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TRANSPARENCY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE PRIORITY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET PRIORITY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE ELEMENTCLASS"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET ELEMENTCLASS PRIMARY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TEMPLATE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TEMPLATE """""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES MAKECOPY OFF"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENTIREELEMENT OFF"

    SetCExpressionValue "tcb->msToolSettings.general.useFence", 0, "CHANGEATTRIBS"

    CadInputQueue.SendCommand "LOCK FENCE INSIDE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LEVEL ""Border-titleblock"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE COLOR"

    point.X = startPoint.X - -34.7268641571481
    point.Y = startPoint.Y + 8.65431517810213
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"
    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendKeyin "level purge all"

    point.X = startPoint.X - -37.7780284085457
    point.Y = startPoint.Y - -2.10891197072276
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -32.9542348920698
    point.Y = startPoint.Y - -3.44201723772157
    point.Z = startPoint.Z + 2.45666666666632
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    point.X = startPoint.X - -31.9062083245021
    point.Y = startPoint.Y - 0.844605732886673
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -27.8145977524912
    point.Y = startPoint.Y + 4.68605728214073
    point.Z = startPoint.Z + 2.45666666666631
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendCommand "NEWFILE P:\Active Projects\PGE\Substation\6446 BELL\2000 Substation\2300 Engineering\2310 Electrical\
2311 Drawings\Indoor\bell7313b0.dgn"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "EDIT SINGLE DIALOG"

    point.X = startPoint.X - -29.4404212279644
    point.Y = startPoint.Y - -1.27945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine BELL-7313"
    point.X = startPoint.X - -31.7404212279644
    point.Y = startPoint.Y - -0.779456644179446
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y - -1.77945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine 21"

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y - -1.77945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -29.7404212279644
    point.Y = startPoint.Y - -3.07945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine BELL SUBSTATION #3                                      "

    point.X = startPoint.X - -29.5404212279644
    point.Y = startPoint.Y + 3.67945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y + 3.57945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine 15KV OUTDOOR SWITCHGEAR                                      "

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y + 3.57945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -30.3404212279644
    point.Y = startPoint.Y - -3.27945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1
    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine UNIT 4 - FDR. BKR R304 CONTROL SCHEM.
         "

    point.X = startPoint.X - -30.3404212279644
    point.Y = startPoint.Y - -3.27945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -31.2404212279644
    point.Y = startPoint.Y - -2.97945664417945
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendKeyin "dialog viewsettings popup"

    CadInputQueue.SendKeyin "MDL KEYIN BENTLEY.VIEWATTRIBUTESDIALOG,VAD VIEWATTRIBUTESDIALOG SETATTRIBUTE 0 DataFields False"

    CadInputQueue.SendKeyin "VBA RUN BUTTONS"

    CadInputQueue.SendCommand "PRINT EXECUTE"

    point.X = startPoint.X - -31.5756002218801
    point.Y = startPoint.Y + 6.28452769205179
    point.Z = startPoint.Z + 2.45666666666623
    CadInputQueue.SendDataPoint point, 1

    SetCExpressionValue "msDialogState.gridInfo.roundoffUnit", (ActiveModelReference.UORsPerMasterUnit * 0.05), "MGDSHOOK"

    CadInputQueue.SendCommand "ACTIVE UNITROUND"

    point.X = startPoint.X - -31.2800461097629
    point.Y = startPoint.Y + 4.80584821066881
    point.Z = startPoint.Z + 2.45666666666623
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    point.X = startPoint.X - -28.8031741484325
    point.Y = startPoint.Y - -2.92696091312891
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -26.650079006342
    point.Y = startPoint.Y - -2.52282284325535
    point.Z = startPoint.Z + 2.45666666666623
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -27.4721049433915
    point.Y = startPoint.Y - -1.33987925814897
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1
    point.X = startPoint.X - -27.439208485695
    point.Y = startPoint.Y - -1.66904616878726
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -27.2683011078185
    point.Y = startPoint.Y - -1.92929375751067
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -27.3094216799392
    point.Y = startPoint.Y - -2.45583223370748
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendKeyin Chr$(27)

    CadInputQueue.SendCommand "MOVE ICON"

    point.X = startPoint.X - -27.1904212279644
    point.Y = startPoint.Y - -2.17945664417945
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -27.8404212279644
    point.Y = startPoint.Y - -1.57945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - -22.7476082103034
    point.Y = startPoint.Y - -0.687594762848514
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -21.366213990625
    point.Y = startPoint.Y + 4.56109209994958
    point.Z = startPoint.Z + 2.45666666666622
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MOVE ICON"

    point.X = startPoint.X - -21.4883140025202
    point.Y = startPoint.Y + 4.29160664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1
    point.X = startPoint.X - -21.4404212279644
    point.Y = startPoint.Y + 5.52945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendCommand "PRINT EXECUTE"

    point.X = startPoint.X - -31.3956977188522
    point.Y = startPoint.Y + 6.1559468675837
    point.Z = startPoint.Z + 2.45666666666623
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -31.3956977188522
    point.Y = startPoint.Y + 6.16880495003051
    point.Z = startPoint.Z + 2.45666666666623
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -31.3956977188522
    point.Y = startPoint.Y + 6.18166303247732
    point.Z = startPoint.Z + 2.45666666666623
    CadInputQueue.SendDataPoint point, 1

'   The following statement opens modal dialog "Open"

    CadInputQueue.SendCommand "DIALOG OPENFILE"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    point.X = startPoint.X - 7.39583659943892E-02
    point.Y = startPoint.Y - 8.62494185738905E-02
    point.Z = startPoint.Z - 6.10623E-15
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.363088088016569
    point.Y = startPoint.Y + 0.207879928787902
    point.Z = startPoint.Z - 6.10623E-15
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"
    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 7.87771946947586E-02
    point.Y = startPoint.Y - 0.115180174052099
    point.Z = startPoint.Z - 6.10623E-15
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.184791426102891
    point.Y = startPoint.Y + 0.101800492034469
    point.Z = startPoint.Z - 6.10623E-15
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "DELETE ELEMENT"

    point.X = startPoint.X - 0.160697282601043
    point.Y = startPoint.Y - 5.24968705159797E-02
    point.Z = startPoint.Z - 6.10623E-15
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.247436199207697
    point.Y = startPoint.Y + 9.21569068750659E-02
    point.Z = startPoint.Z - 6.10623E-15
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "DELETE ELEMENT"

    CadInputQueue.SendCommand "MDL SILENTLOAD USTNVBA MACROS"

'   The following statement opens modal dialog "Color Table"

    CadInputQueue.SendCommand "DIALOG COLOR"

    CadInputQueue.SendKeyin "level purge all"

    CadInputQueue.SendReset

'   The following statement opens modal dialog "Color Table"

    CadInputQueue.SendCommand "DIALOG COLOR"

    CadInputQueue.SendKeyin "level purge all"

    point.X = startPoint.X - 3.87185420612206
    point.Y = startPoint.Y - 0.87389677051904
    point.Z = startPoint.Z - 2.388367E-14
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 3.93067779865587
    point.Y = startPoint.Y - 0.638457679713302
    point.Z = startPoint.Z - 2.388367E-14
    CadInputQueue.SendDataPoint point, 1
    CadInputQueue.SendReset

'   The following statement opens modal dialog "Level/Filter Import"

'   The following statement opens modal dialog "Import Levels"

    CadInputQueue.SendCommand "LEVELMANAGER LIBRARY IMPORT"

    point.X = startPoint.X - 3.61450098878665
    point.Y = startPoint.Y + 2.44358716836557
    point.Z = startPoint.Z - 3.001766E-14
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 3.22374164326869
    point.Y = startPoint.Y - 1.80535267351924
    point.Z = startPoint.Z - 3.001766E-14
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendKeyin Chr$(27)

    CadInputQueue.SendCommand "Change Attributes"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES USEACTIVE ON"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE LEVEL"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LEVEL ""Vendor"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE COLOR"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET COLOR ""0"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE LINESTYLE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET LINESTYLE ""Continuous"""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE WEIGHT"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET WEIGHT 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TRANSPARENCY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TRANSPARENCY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE PRIORITY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET PRIORITY 0"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE ELEMENTCLASS"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET ELEMENTCLASS PRIMARY"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES DISABLE TEMPLATE"
    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES SET TEMPLATE """""

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES MAKECOPY OFF"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENTIREELEMENT OFF"

    SetCExpressionValue "tcb->msToolSettings.general.useFence", 0, "CHANGEATTRIBS"

    CadInputQueue.SendCommand "LOCK FENCE INSIDE"

    CadInputQueue.SendKeyin "CHANGE ATTRIBUTES ENABLE COLOR"

    point.X = startPoint.X + 6.19735445764391E-02
    point.Y = startPoint.Y + 1.48711586196725
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"

    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

    point.X = startPoint.X + 0.135503035243701
    point.Y = startPoint.Y - 0.77530665124414
    point.Z = startPoint.Z - 2.445266E-14
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 3.52258912545257
    point.Y = startPoint.Y + 3.51042054857907
    point.Z = startPoint.Z - 2.445266E-14
    CadInputQueue.SendDataPoint point, 1

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"

    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"
    CadInputQueue.SendCommand "NEWFILE U:\New folder\bellSWGRTITLEBLOCK.dgn"

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    point.X = startPoint.X - -33.6289917232435
    point.Y = startPoint.Y - 4.6783607773943
    point.Z = startPoint.Z + 2.4566666666663
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -21.0670294407541
    point.Y = startPoint.Y + 12.8294198708256
    point.Z = startPoint.Z + 2.4566666666663
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MDL LOAD CLIPBRD COPY"

    CadInputQueue.SendCommand "NEWFILE ""P:\Active Projects\PGE\Substation\6446 BELL\2000 Substation\2300 Engineering\2310 Electrica
l\2311 Drawings\Indoor\bell7313c0.dgn"",""~9308"""

    CadInputQueue.SendKeyin "task sendtaskchangedasync"

    CadInputQueue.SendKeyin "task sendtaskchangedasync ""\Drawing"""

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "COMPONENTVIEW COMPONENTSETOVERRIDE SUSPEND"

    CadInputQueue.SendCommand "MDL KEYIN CLIPBRD CLIPBOARD PASTE"

    point.X = startPoint.X - 37.4892706296046
    point.Y = startPoint.Y + 30.9404508829546
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    point.X = startPoint.X - 5.32835376563225
    point.Y = startPoint.Y + 69.9609485976437
    point.Z = startPoint.Z - -2.45666666666545
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 114.798833963212
    point.Y = startPoint.Y - 44.4094757385139
    point.Z = startPoint.Z - -2.45666666666545
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "SCALE ICON"
    CadInputQueue.SendCommand "ACTIVE XSCALE 0.3900"

    CadInputQueue.SendCommand "ACTIVE SCALE"

    point.X = startPoint.X + 105.834118461059
    point.Y = startPoint.Y + 28.249382075045
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "MOVE ICON"

    CadInputQueue.SendTentativePoint Point3dFromXYZ(98.2705269009586, 38.3111785306598, 2.5399999999987), 1

    point.X = startPoint.X + 95.1037765401523
    point.Y = startPoint.Y + 38.4830111570074
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.8272901125179, 41.5777451127749, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7878628767806, 41.5777451127749, 0#), 1

    CadInputQueue.SendReset

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.6892947874374, 41.5382936268357, 2.5399999999987), 1

    point.X = startPoint.X - 21.1149206296046
    point.Y = startPoint.Y + 41.4404508829546
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "MOVE ICON"

    CadInputQueue.SendTentativePoint Point3dFromXYZ(98.0953691155843, 38.4380474080459, 2.53999999999871), 1

    point.X = startPoint.X + 95.1037765401523
    point.Y = startPoint.Y + 38.4830111570074
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendCommand "LOCK ASSOCIATION OFF"

    CadInputQueue.SendCommand "LOCK UNIT ON"

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7282985463777, 41.402149975998, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7282985463777, 41.402149975998, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7282985463777, 41.402149975998, 0#), 1
    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7899036022172, 41.5254358695581, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7899036022172, 41.5254358695581, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7899036022172, 41.5254358695581, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.7899036022172, 41.5254358695581, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.6520087280777, 41.5632255108319, 0#), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(-19.6520087280777, 41.5632255108319, 0#), 1

    point.X = startPoint.X - 22.6143206296046
    point.Y = startPoint.Y + 41.4404508829546
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    point.X = startPoint.X - 66.0404762882776
    point.Y = startPoint.Y + 46.3408407936845
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 9.49248525517938
    point.Y = startPoint.Y + 6.9533530554312
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "MOVE ICON"

    point.X = startPoint.X - 54.6143206296046
    point.Y = startPoint.Y + 19.9404508829546
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendKeyin "xy=0,0"

    CadInputQueue.SendReset

    CadInputQueue.SendCommand "CHOOSE ELEMENT"

    SetCExpressionValue "powerSelectInfo.prefs.currMode", 4, "PSELECT"

    CadInputQueue.SendCommand "POWERSELECTOR DESELECT"

    CadInputQueue.SendKeyin "dialog viewsettings popup"
    CadInputQueue.SendKeyin "MDL KEYIN BENTLEY.VIEWATTRIBUTESDIALOG,VAD VIEWATTRIBUTESDIALOG SETATTRIBUTE 0 DataFields False"

    CadInputQueue.SendKeyin "dialog viewsettings popup"

    CadInputQueue.SendKeyin "MDL KEYIN BENTLEY.VIEWATTRIBUTESDIALOG,VAD VIEWATTRIBUTESDIALOG SETATTRIBUTE 0 DataFields True"

    CadInputQueue.SendCommand "EDIT SINGLE DIALOG"

    point.X = startPoint.X - -29.4404212279644
    point.Y = startPoint.Y - -2.07945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine BELL-XXXX"

    point.X = startPoint.X - -29.4404212279644
    point.Y = startPoint.Y - -2.07945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine BELL-7314"

    point.X = startPoint.X - -30.9404212279644
    point.Y = startPoint.Y - -1.57945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y - -2.27945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y - -2.17945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine 2 "

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y - -2.17945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -29.9404212279644
    point.Y = startPoint.Y - -2.47945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - -29.6404212279644
    point.Y = startPoint.Y - -3.27945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine BELL SUBSTATION #3                                      "

    point.X = startPoint.X - -29.6404212279644
    point.Y = startPoint.Y - -3.27945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y - -3.27945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine 15KV OUTDOOR SWITCHGEAR                                      "

    point.X = startPoint.X - -30.0404212279644
    point.Y = startPoint.Y - -3.27945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -30.4404212279644
    point.Y = startPoint.Y - -3.17945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine UNIT 5 - MAIN BKR R302 CONTROL SCHEM.
         "

    point.X = startPoint.X - -30.4404212279644
    point.Y = startPoint.Y - -3.17945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - -32.1404212279644
    point.Y = startPoint.Y - -2.37945664417945
    point.Z = startPoint.Z - 0.083333333333315
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendKeyin "dialog viewsettings popup"

    CadInputQueue.SendKeyin "MDL KEYIN BENTLEY.VIEWATTRIBUTESDIALOG,VAD VIEWATTRIBUTESDIALOG SETATTRIBUTE 0 DataFields False"

'   The following statement opens modal dialog "Design File Settings"

    CadInputQueue.SendCommand "MDL SILENTLOAD DGNSET"
    CadInputQueue.SendCommand "FILEDESIGN"

    CadInputQueue.SendCommand "MDL SILENTUNLOAD DGNSET"

'   The following statement opens modal dialog "Color Table"

    CadInputQueue.SendCommand "DIALOG COLOR"

    CadInputQueue.SendCommand "EXIT"

    CadInputQueue.SendCommand "PRINT EXIT PLOTDLG"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
Sub Macro6()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro6ModalHandler
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Print Attributes"

'   Start a command
    CadInputQueue.SendCommand "PRINT ATTRIBDIALOG"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
Sub Macro7()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    CommandState.StartDefaultCommand
End Sub
Sub Macro8()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    Dim modalHandler As New Macro8ModalHandler
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Print Attributes"

'   Start a command
    CadInputQueue.SendCommand "PRINT ATTRIBDIALOG"

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
```

```
End Sub
Sub Macro9()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"

'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.098639241090924, 14.4992497162524, 1.4111111111138), 1

'   Coordinates are in master units
    startPoint.X = 0#
    startPoint.Y = 14.6666666666666
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(22.6630469909838, -4.14684863753751E-03, 1.41111111111402), 1

    point.X = startPoint.X + 22.6666666666667
    point.Y = startPoint.Y - 14.6666666666666
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X + 23.1448617967697
    point.Y = startPoint.Y - 6.5622857142857
    point.Z = startPoint.Z + 1.41111111111123
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "FIT VIEW EXTENDED 5"

    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 21.7245776434224
    point.Y = startPoint.Y - 8.34410349586891
    point.Z = startPoint.Z + 1.41111111111421
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 21.6370483654411
    point.Y = startPoint.Y - 14.7128887948723
    point.Z = startPoint.Z + 1.41111111111426
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub

Sub Macro10()
```

```
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"

'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.365598749417673, 43.5815800805484, 4.23333333333436), 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.102272749327261, 44.0333269768638, 4.23333333333436), 1

'   Coordinates are in master units
    startPoint.X = 0#
    startPoint.Y = 43.9999999999999
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(67.9812786286272, 3.29940720802703E-02, 4.23333333333444), 1

    point.X = startPoint.X + 68#
    point.Y = startPoint.Y - 43.9999999999999
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X + 8.83333333333333
    point.Y = startPoint.Y + 56.1666666666667
    point.Z = startPoint.Z + 4.23333333333333
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "FIT VIEW EXTENDED 5"

    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 65.6749694500775
    point.Y = startPoint.Y - 24.7501961741185
    point.Z = startPoint.Z + 4.23333333333444
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 65.7208898602495
    point.Y = startPoint.Y - 44.117079717327
    point.Z = startPoint.Z + 4.23333333333444
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub
Sub Macro11()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Coordinates are in master units
    startPoint.X = -2.38719521710856
    startPoint.Y = 5.40902255639098
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 5

'   Send a keyin that can be a command string
    CadInputQueue.SendKeyin "dialog viewsettings popup"

    CadInputQueue.SendKeyin "MDL KEYIN BENTLEY.VIEWATTRIBUTESDIALOG,VAD VIEWATTRIBUTESDIALOG SETATTRIBUTE 4 DataFields False"

    CommandState.StartDefaultCommand
End Sub
Sub Macro12()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"

'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(0.493826490298015, 30.0246028833885, 0#), 1

'   Coordinates are in master units
    startPoint.X = 0#
    startPoint.Y = 30#
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(41.97131768619, 1.17293206833481E-02, 0#), 1

    point.X = startPoint.X + 42.0000000000001
    point.Y = startPoint.Y - 30#
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X + 1.79290742839691
    point.Y = startPoint.Y + 3.46071529917275
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "PRINT MAXIMIZE"

    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 40.9646372074583
    point.Y = startPoint.Y - 30.3439460111004
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 40.801902248909
    point.Y = startPoint.Y - 20.3922465283965
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub
Sub Macro13()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"

'   Send a tentative point
'   Coordinates are in master units
    CadInputQueue.SendTentativePoint Point3dFromXYZ(-7.68330441137402, 29.5718401832303, 0#), 1

'   Coordinates are in master units
    startPoint.X = -8.00000000000005
    startPoint.Y = 30#
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    CadInputQueue.SendTentativePoint Point3dFromXYZ(33.9605377145485, 2.06433146363044E-02, 0#), 1

    point.X = startPoint.X + 42#
    point.Y = startPoint.Y - 30#
    point.Z = startPoint.Z
    CadInputQueue.SendAdjustedDataPoint point, 1

    point.X = startPoint.X - 4.59999999999996
    point.Y = startPoint.Y - 14.05
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 5

    CadInputQueue.SendCommand "FIT VIEW EXTENDED 5"
    CadInputQueue.SendCommand "WINDOW AREA EXTENDED 1"

    point.X = startPoint.X + 40.7100968867561
    point.Y = startPoint.Y - 20.425106943707
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 40.9609209462061
    point.Y = startPoint.Y - 30.0283192740974
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub
Sub Macro14()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Coordinates are in master units
    startPoint.X = -6.56545142075177
    startPoint.Y = 54.3998934213841
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 5.95906729282748
    point.Y = startPoint.Y + 4.23707626929689
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

'   Start a command
    CadInputQueue.SendCommand "PLACE FENCE ICON"

    point.X = startPoint.X - 0.439089168945145
    point.Y = startPoint.Y + 8.69041997317247
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 6.20997538936757
    point.Y = startPoint.Y + 13.4610539948993
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendCommand "PLACE FENCE ICON"

    point.X = startPoint.X + 0.354799730263835
    point.Y = startPoint.Y - 3.12043012749257
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 5.91790268323876
    point.Y = startPoint.Y + 4.72600740824241
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub
Sub Macro15()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Coordinates are in master units
    startPoint.X = -7.0696445597244
    startPoint.Y = 57.3839304543931
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 6.42324727142667
    point.Y = startPoint.Y + 6.62867043018892
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

'   Start a command
    CadInputQueue.SendCommand "ORDER ELEMENT FRONT"

    CommandState.StartDefaultCommand
End Sub
Sub Macro16()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Coordinates are in master units
    startPoint.X = 33.1527004748686
    startPoint.Y = 0.754570801973472
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

'   Send a message string to an application
'   Content is defined by the application
    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 16"
    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 16"

    point.X = startPoint.X + 2.02202552933635
    point.Y = startPoint.Y + 0.511180800000016
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.474140430326564
    point.Y = startPoint.Y + 11.9115087470084
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 27"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 27"

    point.X = startPoint.X + 0.491130115385928
    point.Y = startPoint.Y + 11.7356298362372
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.531604437698213
    point.Y = startPoint.Y + 12.735398608026
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 83"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 83"

    point.X = startPoint.X + 1.09579141498126
    point.Y = startPoint.Y + 13.787661622426
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.486300933930082
    point.Y = startPoint.Y + 18.8328585963654
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    point.X = startPoint.X + 0.966672293459965
    point.Y = startPoint.Y + 19.4093287712902
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.483975647140291
    point.Y = startPoint.Y + 19.2116423764102
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    point.X = startPoint.X - 0.465645794156544
    point.Y = startPoint.Y + 20.0158674528391
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    point.X = startPoint.X - 0.427927708092867
    point.Y = startPoint.Y + 19.6215636852625
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    point.X = startPoint.X - 0.435867267566309
    point.Y = startPoint.Y + 20.4314206033249
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 setColor 3"

    CadInputQueue.SendMessageToApplication "WORDPROC", "1 selection 13 18"

    point.X = startPoint.X - 0.274746316731445
    point.Y = startPoint.Y + 20.4222726665818
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
```

```
End Sub
Sub Macro17()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Coordinates are in master units
    startPoint.X = 32.5440839946826
    startPoint.Y = 13.6218545103832
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

'   Send a message string to an application
'   Content is defined by the application
    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine REVISED PRIOR TO CONSTRUCTION, AWO 1000001215.
                                                "

    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.857899547846586
    point.Y = startPoint.Y - 0.754601892977787
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1, 2

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X - 8.19649249535317E-02
    point.Y = startPoint.Y + 6.9992059638519
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 9.84945181524211E-02
    point.Y = startPoint.Y + 5.95595517648597
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine PES   "

    point.X = startPoint.X + 9.84945181524211E-02
    point.Y = startPoint.Y + 5.95595517648597
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X + 8.10086674956736E-02
    point.Y = startPoint.Y + 6.33216249704301
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine DDB   "

    point.X = startPoint.X + 8.10086674956736E-02
    point.Y = startPoint.Y + 6.33216249704301
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X + 5.91513541747446E-02
    point.Y = startPoint.Y + 6.75648935860153
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine RCL   "

    point.X = startPoint.X + 5.91513541747446E-02
    point.Y = startPoint.Y + 6.74774035114672
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X + 7.22657421673034E-02
    point.Y = startPoint.Y + 7.1501946940682
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine DDB   "

    point.X = startPoint.X + 7.22657421673034E-02
    point.Y = startPoint.Y + 7.1501946940682
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    point.X = startPoint.X + 8.53801301598622E-02
    point.Y = startPoint.Y + 7.52202751089783
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine REJ   "

    point.X = startPoint.X + 8.53801301598622E-02
    point.Y = startPoint.Y + 7.52202751089783
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "
    point.X = startPoint.X - 0.771426552020685
    point.Y = startPoint.Y + 7.2114377462519
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CommandState.StartDefaultCommand
End Sub
```

```
Sub Buttons()
  frmButtons.show vbModeless
End Sub


Sub D_BORDER()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "DIALOG CELLMAINTENANCE"

    Dim modalHandler As New Macro1ModalHandler3
    AddModalDialogEventsHandler modalHandler

'   The following statement opens modal dialog "Attach Cell Library"

    CadInputQueue.SendCommand "ATTACH LIBRARY"

'   Set a variable associated with a dialog box
    SetCExpressionValue "tcb->activeCell", "BDR-D10", ""

'   Send a keyin that can be a command string
    CadInputQueue.SendKeyin "inputmanager currenttask"

    CadInputQueue.SendCommand "INPUTMANAGER MENU -609 2"

    CadInputQueue.SendCommand "DMSG ACTIVATETOOLBYPATH \Drawing\Cells\Place Active Cell"

    CadInputQueue.SendCommand "PLACE CELL ICON"

    CadInputQueue.SendKeyin "xy=0,0"

'   Send a reset to the current command
    CadInputQueue.SendReset

    RemoveModalDialogEventsHandler modalHandler
    CommandState.StartDefaultCommand
End Sub
Sub CRTS()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Start a command
    CadInputQueue.SendCommand "INPUTMANAGER MENU -609,7"

    CadInputQueue.SendCommand "DMSG ACTIVATETOOLBYPATH \Drawing\Text\Edit Text"

    CadInputQueue.SendCommand "EDIT TEXT"

'   Coordinates are in master units
    startPoint.X = 32.143094
    startPoint.Y = 4.92251
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

'   Send a message string to an application
'   Content is defined by the application
    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine CURTIS SUBSTATION"

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "NextLine "

    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine CURTIS SUBSTATION"

    point.X = startPoint.X + 1.191537
    point.Y = startPoint.Y - 2.839316
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine D-<<     >>"

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "NextLine "

'   Send a keyin that can be a command string
    CadInputQueue.SendKeyin Chr$(27)

    CadInputQueue.SendCommand "INPUTMANAGER MENU -609,7"

    CadInputQueue.SendCommand "DMSG ACTIVATETOOLBYPATH \Drawing\Text\Fill In Single Enter-Data Field"

    CadInputQueue.SendCommand "EDIT SINGLE DIALOG"

    point.X = startPoint.X + 1.173366
    point.Y = startPoint.Y - 2.799316
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine CON     "

    point.X = startPoint.X + 1.173366
    point.Y = startPoint.Y - 2.799316
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "
    point.X = startPoint.X + 0.144069
    point.Y = startPoint.Y + 10.062997
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine AWO 0000026594.
 "



    point.X = startPoint.X - 0.242137
    point.Y = startPoint.Y + 16.100209
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.268499
    point.Y = startPoint.Y - 3.622898
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine CRTS-         "

    point.X = startPoint.X + 0.268499
    point.Y = startPoint.Y - 3.622898
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    CadInputQueue.SendMessageToApplication "TEXTEDIT", "FirstLine "

    CommandState.StartDefaultCommand
End Sub



Sub Macro1()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

'   Set a variable associated with a dialog box
    SetCExpressionValue "plotUI.uiPlotArea", 2, "PLOTDLG"

'   Coordinates are in master units
    startPoint.X = 35.175694
    startPoint.Y = 4.473955
    startPoint.Z = 0#

'   Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.463557
    point.Y = startPoint.Y - 0.42816
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

'   Start a command
    CadInputQueue.SendCommand "MDL SILENTLOAD USTNVBA IDE"

    SetCExpressionValue "plotUI.uiPlotArea", 3, "PLOTDLG"

'   Send a keyin that can be a command string
    CadInputQueue.SendKeyin "VBA RUN BUTTONS"

    point.X = startPoint.X + 2.448534
    point.Y = startPoint.Y + 1.35584
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 0.237722
    point.Y = startPoint.Y + 1.28448
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 2.115724
    point.Y = startPoint.Y + 2.652213
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X - 0.33281
    point.Y = startPoint.Y + 1.71264
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1
End Sub
```

Resolution - 1

```
Option Explicit
Option Base 1
Sub test_getRes()
Dim msg As String

msg = GetResolution
MsgBox msg


End Sub

Function GetResolution() As String
 On Error GoTo errhnd
 Dim lngDspPrty As Long
 Dim retMsg As String
 Dim Resl As String
 Resl = ""
 Dim ele As CellElement
 Dim rng As Range3d
 Dim BorderName As String
 Dim msg As String
 msg = ""
 Set ele = GetBorder(True, True)
 rng = ele.Range
 BorderName = ele.Name

 '****************************************
 Dim DeltaY As Long
 DeltaY = rng.High.Y - rng.Low.Y

 Select Case BorderName
     Case "BDR-D10"
         'if deltaY   - check if 10000
         If DeltaY <> 22 Then
            msg = "Border Size or Resolution Error!"
           Else
            Resl = "10000"
         End If
     Case "BDR-E10"
         'if deltaY   - check if 10000
         If DeltaY <> 30 Then
            msg = "Border Size or Resolution Error!"
           Else
            Resl = "10000"
         End If
     Case "BDR-D12"
        'if deltaY is not when one of the following
        'send out working resolution error
        Select Case DeltaY
          Case 352
             '1/16"=1'-0"
             Resl = "750"
          Case 235
             '3/32"=1'-0"
             Resl = "1125"
          Case 176
             '1/8"=1'-0"
             Resl = "1500"
          Case 117
             '3/16"=1'-0"
             Resl = "2250"
          Case 88
             '1/4"=1'-0"
             Resl = "3000"
          Case 59
             '3/8"=1'-0"
             Resl = "4500"
          Case 44
             '1/2"=1'-0"
             Resl = "6000"
          Case 29
             '3/4"=1'-0"
             Resl = "9000"
          Case 22
             '1"=1'-0"
             Resl = "12000"
          Case 15
             '1 1/2"=1'-0"
             Resl = "18000"
          Case 11
             '2"=1'-0"
             Resl = "24000"
          Case 7
             '3"=1'-0"
             Resl = "36000"
          Case Else
             'Working Resolution is incorrect
             msg = "Border Size or Resolution Error!"
          End Select

     Case "BDR-E12"
          'if deltaY is not when one of the following
          'send out working resolution error
           Select Case DeltaY
          Case 480
             '1/16"=1'-0"
             Resl = "750"
          Case 320
             '3/32"=1'-0"
             Resl = "1125"
          Case 240
             '1/8"=1'-0"
             Resl = "1500"
          Case 160
             '3/16"=1'-0"
             Resl = "2250"
          Case 120
             '1/4"=1'-0"
             Resl = "3000"
          Case 80
             '3/8"=1'-0"
             Resl = "4500"
          Case 60
             '1/2"=1'-0"
             Resl = "6000"
          Case 40
             '3/4"=1'-0"
             Resl = "9000"
          Case 30
             '1"=1'-0"
             Resl = "12000"
          Case 20
             '1 1/2"=1'-0"
             Resl = "18000"
          Case 15
             '2"=1'-0"
             Resl = "24000"
          Case 10
             '3"=1'-0"
             Resl = "36000"
          Case Else
             'Working Resolution is incorrect
             msg = "Border Size or Resolution Error!"
          End Select
     Case Else
       'Not expecting to have no Border, but Cell name could have been changed
       msg = "Border Cell Name is Unexpected!"
End Select

Select Case Resl
        'Set General Text Height according drawing scale
          Case "10000"
             'Non-Scaled
             ActiveSettings.TextStyle.Height = 0.1
             ActiveSettings.TextStyle.Width = 0.1
          Case "750"
             ActiveSettings.TextStyle.Height = 1.60026
             ActiveSettings.TextStyle.Width = 1.60026
          Case "1125"
             ActiveSettings.TextStyle.Height = 1.06640625
             ActiveSettings.TextStyle.Width = 1.06640625
          Case "1500"
             ActiveSettings.TextStyle.Height = 0.799479
             ActiveSettings.TextStyle.Width = 0.799479
          Case "2250"
             ActiveSettings.TextStyle.Height = 0.533854
             ActiveSettings.TextStyle.Width = 0.533854
          Case "3000"
             ActiveSettings.TextStyle.Height = 0.399739
             ActiveSettings.TextStyle.Width = 0.399739
          Case "4500"
             ActiveSettings.TextStyle.Height = 0.266927
             ActiveSettings.TextStyle.Width = 0.266927
          Case "6000"
             ActiveSettings.TextStyle.Height = 0.200521
             ActiveSettings.TextStyle.Width = 0.200521
          Case "9000"
             ActiveSettings.TextStyle.Height = 0.1328125
             ActiveSettings.TextStyle.Width = 0.1328125
          Case "12000"
             ActiveSettings.TextStyle.Height = 0.100260416
             ActiveSettings.TextStyle.Width = 0.100260416
          Case "18000"
             ActiveSettings.TextStyle.Height = 0.06640625
             ActiveSettings.TextStyle.Width = 0.06640625
          Case "24000"
             ActiveSettings.TextStyle.Height = 0.0494792
             ActiveSettings.TextStyle.Width = 0.0494792
          Case "36000"
             ActiveSettings.TextStyle.Height = 0.03385416
             ActiveSettings.TextStyle.Width = 0.03385416
          Case Else
             'do nothing
 End Select

  If msg = "" Then
     retMsg = "(dwg resolution: " & Resl & ")"
     GetResolution = retMsg
    Else
     GetResolution = msg
   End If

  Exit Function
errhnd:
   Select Case Err.number
     Case 91 'Get Border didn't find any Borders
             'Could be a raster file a raster Title Block
       MsgBox "Program ended! No title block on this drawing."
       Err.Clear
     End Select
End Function


Function GetBorder(ignoreT As Boolean, BorderExist As Boolean) As Element

  Dim number As Double


  Dim rngBDR As Range3d
  Dim pntBDRs As Point3d
  Dim pntBDRe As Point3d
  Dim rngTBDR As Range3d
  Dim pntTBDRs As Point3d
  Dim pntTBDRe As Point3d
  Dim dblScale As Double
```

Resolution - 5

```
  Dim BorderName As String
  Dim oElem As Element
  Dim oCellElem As CellElement
  Dim BdrObject As CellElement
  Dim TbdrObject As CellElement
  Dim oEnum As ElementEnumerator
  Dim ElementCounter As Long
  Dim BorderType As String

  Dim BorderD10 As Boolean
  Dim BorderE10 As Boolean
  Dim BorderT10 As Boolean

  Dim BorderD12 As Boolean
  Dim BorderE12 As Boolean
  Dim BorderT12 As Boolean

  BorderD10 = False
  BorderE10 = False
  BorderT10 = False
  BorderD12 = False
  BorderE12 = False
  BorderT12 = False

  Set oEnum = ActiveModelReference.Scan()

  While oEnum.MoveNext
        ElementCounter = ElementCounter + 1
        Set oElem = oEnum.Current

        If oElem.IsCellElement Then
          Set oCellElem = oElem

          Select Case oCellElem.Name
            Case "BDR-D10"
              'MsgBox "D10"
              Set BdrObject = oCellElem
              BorderD10 = True
            Case "BDR-D12"
              'MsgBox "D12"
              Set BdrObject = oCellElem
              BorderD12 = True
            Case "BDR-E10"
              'MsgBox "E10"
              Set BdrObject = oCellElem
              BorderE10 = True
            Case "BDR-E12"
              'MsgBox "E12"
              Set BdrObject = oCellElem
              BorderE12 = True
            Case "BDR-T10"
              'MsgBox "T10"
              Set TbdrObject = oCellElem
              BorderT10 = True
            Case "BDR-T12"
              'MsgBox "T12"
              Set TbdrObject = oCellElem
              BorderT12 = True
            Case Else
              'No border found - send by reference this value to avoid error
              BorderExist = False
           End Select
        End If
   Wend

   If ignoreT = True Then
     'MsgBox "ignore T"
       If BorderE10 Or BorderD10 Or BorderD12 Or BorderE12 Then
          Set GetBorder = BdrObject
          BorderExist = True
          Exit Function
         Else
          MsgBox "No D or E Borders in this file"
       End If
   End If

   If ignoreT = False Then
       If BorderT10 Or BorderT12 Then
          BorderExist = True
          Set GetBorder = TbdrObject
          'MsgBox "T border takes priority"
         ElseIf BorderE10 Or BorderD10 Or BorderD12 Or BorderE12 Then
          BorderExist = True
          Set GetBorder = BdrObject
          Exit Function
         Else
          MsgBox "No D or E Borders in this file"
       End If
   End If
End Function
```

```
Option Explicit

Sub PrintInfo(eleComponent As Element)
    With eleComponent
        Debug.Print "Element " & DLongToString(.ID) & " color: " & .color
    End With
End Sub


'
'   This example reloads the ComplexElement from the design file so it has a copy
'   of the element that has the changes that were written to the file.
'
Sub SubElements(eleCell As CellElement)
    Dim eleComponent As Element
    Dim ee As ElementEnumerator
    Dim originalColor As Integer

    '
    ' Get the sub elements and then get the first element
    '

    Set ee = eleCell.GetSubElements
    ee.MoveNext
    'ee.BuildArrayFromContents

    Set eleComponent = ee.Current

    '
    '   Get the color
    '
    originalColor = eleComponent.color
    Debug.Print "The original color is " & originalColor

    '
    '   Change the color and save the changed element to the design file.
    '
    eleComponent.color = originalColor + 1
    eleComponent.Rewrite

    Dim ele As Element
    Dim oModelReference As ModelReference

    '
    '   Now use the element's ModelReference and element ID to reload
    '   the ComplexElement from the design file. The first step
    '   assigns the ComplexElement to a variable of type Element so the program
    '   can use Element's methods.  After setting ele to eleCell, ele and
    '   eleCell refer to the same object but these variables provide access to
    '   different methods and properties.
    '
    Set ele = eleCell
    Set oModelReference = ele.ModelReference
    Set eleCell = oModelReference.GetElementByID(ele.ID)
    '
    '   Get a new enumerator.  Since the program read the changes from the
    '   DesignFile, the complex element now contains the changed component.
    '
    Set ee = eleCell.GetSubElements
    ee.MoveNext
    Set eleComponent = ee.Current

    '
    '   Now it prints the updated color
    '
    PrintInfo eleComponent



End Sub

Function GetBorder()

  Dim oElem As Element
  Dim oCellElem As CellElement
  Dim allElem As ElementEnumerator



  Dim BdrObject As CellElement
  Dim TbdrObject As CellElement
  Dim oEnum As ElementEnumerator
  Dim ElementCounter As Long
  Dim BorderType As String

  Set oEnum = ActiveModelReference.Scan()

  While oEnum.MoveNext
        Set oElem = oEnum.Current

        If oElem.IsCellElement Then
          Set oCellElem = oElem

          Select Case oCellElem.Name
            Case "BDR-D10"
              'MsgBox "D10"
              MsgBox "found border"

                    Dim eleComponent As Element
                    Dim ee As ElementEnumerator
                    Dim originalColor As Integer

                    '
                    ' Get the sub elements and then get the first element
                    '
                    Set ee = oCellElem.GetSubElements
                    ee.MoveNext
                    Set eleComponent = ee.Current

                    '
                    '   Get the color
                    '
                    MsgBox eleComponent.IsGraphical
                    MsgBox eleComponent.IsTextElement


                    'If eleComponent.IsTextElement Then
                    'eleComponent.AsTextElement

                    originalColor = eleComponent.color
                    Debug.Print "The original color is " & originalColor

                    '
                    '   Change the color and save the changed element to the design file.
                    '
                    eleComponent.color = originalColor + 1
                    eleComponent.Rewrite

                     Dim ele As Element
                        Dim oModelReference As ModelReference

                        '
                        '   Now use the element's ModelReference and element ID to reload
                        '   the ComplexElement from the design file. The first step
                        '   assigns the ComplexElement to a variable of type Element so the program
                        '   can use Element's methods.  After setting ele to oCellElement, ele and
                        '   eleCell refer to the same object but these variables provide access to
                        '   different methods and properties.
                        '
                        Set ele = oCellElem
                        Set oModelReference = ele.ModelReference
                        Set oCellElem = oModelReference.GetElementByID(ele.ID)

                        '
                        '   Get a new enumerator.  Since the program read the changes from the
                        '   DesignFile, the complex element now contains the changed component.
                        '
                        Set ee = oCellElem.GetSubElements
                        ee.MoveNext
                        Set eleComponent = ee.Current



                        '
                        '   Now it prints the updated color
                        '
                        PrintInfo eleComponent

            Case Else
              'MsgBox "didn't find a border"
```

```
           End Select
        End If
   Wend

End Function
```

```
Sub settextheight()
  'ActiveSettings.TextStyle.Height = 0.75
  'ActiveSettings.TextStyle.Width = 0.75
  SetCExpressionValue "cloudParams.radius", (ActiveModelReference.UORsPerMasterUnit * ActiveSettings.TextStyle.Height), "COMPCURV"
End Sub

  Sub testTxtHgt()
     MsgBox ActiveSettings.TextStyle.Height
     ActiveSettings.TextStyle.Height = 0.2
     ActiveSettings.TextStyle.Width = 0.2
     MsgBox ActiveSettings.TextStyle.Height
     ActiveSettings.TextStyle.Height = 0.533854
     ActiveSettings.TextStyle.Width = 0.533854
     MsgBox ActiveSettings.TextStyle.Height
  End Sub


Sub test_flipcase()
  Dim UserIn As String, Num As Long
  UserIn = InputBox(Prompt:="Enter some text:", _
                    Title:="FlipCase Test")
   Num = 5
   MsgBox flipcase(UserIn, Num)
   MsgBox UserIn
   MsgBox flipcase(UserIn)
   MsgBox UserIn

End Sub

Function flipcase(tStr As String, Optional nChar) As String
  Dim k As Long

  Dim testC As String * 1  'length of string as 1
   If IsMissing(nChar) Then
     nChar = Len(tStr)
   End If

  For k = 1 To nChar
  testC = Mid(tStr, k, 1)
   If (StrComp(testC, "A", vbBinaryCompare) >= 0) And _
       (StrComp(testC, "Z", vbBinaryCompare) <= 0) Then
       Mid(tStr, k, 1) = UCase(testC)
   ElseIf (StrComp(testC, "a", vbBinaryCompare) >= 0) And _
          (StrComp(testC, "z", vbBinaryCompare) <= 0) Then
       Mid(tStr, k, 1) = UCase(testC)
   End If
   Next k
   flipcase = tStr
End Function

Sub zoomExtents()
   Dim oView As View
   Dim pnt(1 To 4) As Point3d
   Dim intview As Integer
   intview = 1

   CadInputQueue.SendCommand "FIT VIEW EXTENDED 1"
   Set oView = ActiveDesignFile.Views(intview)

   pnt(1).X = oView.Origin.X
   pnt(1).Y = oView.Origin.Y
   pnt(2).X = oView.Extents.X
   pnt(2).Y = oView.Origin.Y
   pnt(3).X = oView.Extents.X
   pnt(3).Y = oView.Extents.Y
   pnt(4).X = oView.Origin.X
   pnt(4).Y = oView.Extents.Y

   pnt1 = oView.Extents
   oView.Redraw
   ActiveDesignFile.Fence.DefineFromModelPoints 1, pnt()
   ActiveDesignFile.Fence.Draw msdDrawingModeHilite

End Sub

Sub TestGetFiles()
   ' Call to test GetFiles function.
   Dim PPath As String
   Dim BellPath As String
   Dim dctDict As Dictionary
   Dim varItem As Variant
   Dim GetTempDir As String
   'GetTempDir = "C:\Users\knowles_keith\Desktop\Microstation_test_Folder"
   PPath = "P:\Active Projects\PGE\Substation\"
   BellPath = "6446 BELL\2000 Substation\2300 Engineering\2310 Electrical\2311 Drawings\Indoor"
   GetTempDir = PPath & BellPath
   'Create new dictionary.
   Set dctDict = New Dictionary
   ' Call recursively, return files into Dictionary object.
   If GetFiles(GetTempDir, dctDict, False) Then
      ' Print items in dictionary.
      For Each varItem In dctDict
         Debug.Print varItem
      Next
   End If
End Sub
Function GetFiles(strPath As String, _
                dctDict As Dictionary, _
                Optional blnRecursive As Boolean) As Boolean

   ' This procedure returns all the files in a directory into
   ' a Dictionary object. If called recursively, it also returns
   ' all files in subfolders.

   Dim fsoSysObj      As FileSystemObject
   Dim fdrFolder      As Folder
   Dim fdrSubFolder   As Folder
   Dim filFile        As File
   ' Return new FileSystemObject.
   Set fsoSysObj = New FileSystemObject

   On Error Resume Next
   ' Get folder.
   Set fdrFolder = fsoSysObj.GetFolder(strPath)
   If Err <> 0 Then
      ' Incorrect path.
      GetFiles = False
      GoTo GetFiles_End
   End If
   On Error GoTo 0

   ' Loop through Files collection, adding to dictionary.
   For Each filFile In fdrFolder.Files
      dctDict.Add filFile.path, filFile.path
   Next filFile

   ' If Recursive flag is true, call recursively.
   If blnRecursive Then
      For Each fdrSubFolder In fdrFolder.SubFolders
         GetFiles fdrSubFolder.path, dctDict, True
      Next fdrSubFolder
   End If

   ' Return True if no error occurred.
   GetFiles = True

GetFiles_End:
   Exit Function
End Function


Sub KJK()
 Dim ob As Application


  Application.ActiveDesignFile.TotalEditingTime


End Sub

Sub testScanFilter()
  Dim rng As Range3d
  Dim pnt3D As Point3d

  Dim mycell As CellInformation
  Dim myCellEnum As CellInformationEnumerator

  Dim myElem As Element
  Dim myEnum As ElementEnumerator
  Dim myFilter As New ElementScanCriteria
  Dim ElementCounter As Long
  Dim myCollection As New Collection
  'myFilter.ExcludeAllTypes
  myFilter.ExcludeAllLevels
  'myFilter.ExcludeAllColors
  'myFilter.IncludeType msdElementTypeText
  'myFilter.IncludeType msdElementTypeTextNode

  myFilter.IncludeLevel ActiveDesignFile.Levels("Border-titleblock")
  myFilter.IncludeLevel ActiveDesignFile.Levels("Border and Titleblock")
  'myFilter.IncludeLevel ActiveDesignFile.Levels("Level 1")
  'myFilter.IncludeLevel ActiveDesignFile.Levels("Existing")

  'myFilter.IncludeOnlyCell "BDR-D10"
  'myFilter.IncludeColor 4
  Set myEnum = ActiveModelReference.Scan(myFilter)

  While myEnum.MoveNext
    ElementCounter = ElementCounter + 1
    Set myElem = myEnum.Current
    myCollection.Add myElem
    MsgBox myElem.AsCellElement.Name & " " & "origin: " & vbLf & _
                  myElem.AsCellElement.Origin.X & ", " & myElem.AsCellElement.Origin.Y

    'MsgBox myElem.AsCellElement.Origin.x & " " & myElem.AsCellElement.Origin.Y
    'MsgBox myElem.AsCellElement.IsGraphical
    rng = myElem.AsCellElement.Range

    'MsgBox "x: " & Str(pnt3D.x = rng.High.x)
    pnt3D.X = rng.High.X
    pnt3D.Y = rng.High.Y
    MsgBox "High X: " & Str(pnt3D.X) & "High Y: " & Str(pnt3D.Y)
     pnt3D.X = rng.Low.X
    pnt3D.Y = rng.Low.Y
    MsgBox "Low X: " & Str(pnt3D.X) & "Low Y: " & Str(pnt3D.Y)
  Wend
  MsgBox ElementCounter & " elements found."
End Sub

Sub SummarizeMessageCenter()
    Dim oMC As MessageCenter
    Dim iMsg As Integer
    Dim limit As Integer

    Set oMC = MessageCenter
    limit = oMC.MessageCount - 1

    For iMsg = 0 To limit
        Dim tMsg As MessageCenterMessage

        tMsg = oMC.GetMessage(iMsg)
        Debug.Print "Message # " & iMsg & ", Priority " & tMsg.Priority & " Contents: " & tMsg.MESSAGE
        If tMsg.Details <> "" Then
            Debug.Print "---------Start of Details---------"
            Debug.Print tMsg.Details
            Debug.Print "---------End of Details-----------"
        End If
    Next
End Sub

Sub test_message()
   Dim DetMessage As String
   Dim dash As String
   dash = " ---- "
   DetMessage = "[" & "LONG STRING" & dash & _
                "NEXT ERROR" & dash & _
                "2 line" & dash & _
                 "2 line" & dash & _
                 "2 line" & dash & _
                "2 line" & dash & _
                "2 line" & dash & _
                 "2 line" & dash & _
                  "2 line" & dash & _
                 "2 line" & dash & _
                 "2 line" & dash & _
                "2 line" & dash & _
                "2 line" & dash & _
                 "2 line" & "]"

   ShowTempMessage msdStatusBarAreaLeft, "GOOD" & DetMessage
   ShowTempMessage msdStatusBarAreaMiddle, "ERRORS", DetMessage
End Sub

Sub AddInfoMessage()
    Dim msg As String
    msg = "Message type " & msdMessageCenterPriorityInfo & " using all of the defaults"
    MessageCenter.AddMessage msg
End Sub
Sub AddWarningMessage()
    Dim msg As String
    msg = "WARNING MESSAGE is type " & msdMessageCenterPriorityWarning

    MessageCenter.AddMessage msg, Priority:=msdMessageCenterPriorityWarning
End Sub
Sub AddErrorMessage()
    Dim msg As String
    msg = "ERROR MESSAGE is type " & msdMessageCenterPriorityError

    MessageCenter.AddMessage msg, "This is the detail information for the error message", msdMessageCenterPriorityError, True
End Sub
Sub AddDebugMessage()
    Dim msg As String
    msg = "DEBUG MESSAGE is type " & msdMessageCenterPriorityDebug

    MessageCenter.AddMessage msg
End Sub
Sub AddPriorityNone()
    Dim msg As String
    msg = "Message type " & msdMessageCenterPriorityNone
    MessageCenter.AddMessage msg
End Sub
```

```
Dim WithEvents myMS As Application

Private Sub Class_Initialize()

End Sub

Private Sub myMS_OnDesignFileOpened(ByVal DesignFileName As String)

Buttons


End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGE_PikeDrafterCells.CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Preferences [irasb]" Then

    '   Set a variable associated with a dialog box
        SetCExpressionValue "savePrefs.textEditorStyle", 4, "USERPREF"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Preferences [irasb]

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Preferences [irasb]" Then

    '   Set a variable associated with a dialog box
        SetCExpressionValue "savePrefs.textEditorStyle", 0, "USERPREF"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Preferences [irasb]

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Import Levels" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd Electrical.levels.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Import Levels

    If DialogBoxName = "Level/Filter Import" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Level/Filter Import

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGESHEET(new).CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGE_PikeDrafterCells.CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Preferences [descartes]" Then

    '   Set a variable associated with a dialog box
        SetCExpressionValue "savePrefs.textEditorStyle", 0, "USERPREF"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Preferences [descartes]

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGE_PikeDrafterCells.CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Preferences [descartes]" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Preferences [descartes]

    If DialogBoxName = "Open" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd P:\Active Projects\PGE\Substation\6454 Rivergate II
\2000 Substation\2300 Engineering\2310 Electrical\2311 Drawings\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd rvgt6603c1.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Open

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Preferences [descartes]" Then

    '   Set a variable associated with a dialog box
        SetCExpressionValue "savePrefs.textEditorStyle", 4, "USERPREF"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Preferences [descartes]

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Print - Raster Options" Then

    '   Set a variable associated with a dialog box
        SetCExpressionValue "rasterOptionsUI.quality", 100, "PLOTDLG"
    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Print - Raster Options

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGE_PikeDrafterCells.CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Color Table" Then

        CadInputQueue.SendCommand "CT= "

    End If  ' Color Table

    If DialogBoxName = "Design File Settings" Then

    '   The following statement opens modal dialog "Advanced Unit Settings"

    '   The following statement opens modal dialog "Alert"

    '   Set a variable associated with a dialog box
    '   This only modifies a few bits of the variable it changes. It first
    '   creates a mask for clearing the bits it will change. Then it gets
    '   the variable and uses the mask to clear those bits. Finally
    '   it sets the desired bits in the value and saves the updated value.
        lngTemp = Not 3
        lngTemp = GetCExpressionValue("dgnSet.unitFormatDGN", "DGNSET") And lngTemp
        SetCExpressionValue "dgnSet.unitFormatDGN", lngTemp Or 1, "DGNSET"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Alert" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Alert

    If DialogBoxName = "Advanced Unit Settings" Then

        SetCExpressionValue "dgnSet.adv_uorPerStorage", 10000, "DGNSET"
    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Advanced Unit Settings

    If DialogBoxName = "Import Levels" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd Electrical.levels.dgn"
    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Import Levels

    If DialogBoxName = "Level/Filter Import" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Level/Filter Import

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Design File Settings" Then

    '   The following statement opens modal dialog "Advanced Unit Settings"

    '   The following statement opens modal dialog "Alert"

    '   Set a variable associated with a dialog box
    '   This only modifies a few bits of the variable it changes. It first
    '   creates a mask for clearing the bits it will change. Then it gets
    '   the variable and uses the mask to clear those bits. Finally
    '   it sets the desired bits in the value and saves the updated value.
        lngTemp = Not 3
        lngTemp = GetCExpressionValue("dgnSet.unitFormatDGN", "DGNSET") And lngTemp
        SetCExpressionValue "dgnSet.unitFormatDGN", lngTemp Or 1, "DGNSET"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Alert" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Alert

    If DialogBoxName = "Advanced Unit Settings" Then

        SetCExpressionValue "dgnSet.adv_uorPerStorage", 10000, "DGNSET"
    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Advanced Unit Settings

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGESCH.CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGESHEET(new).CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Color Table" Then

        CadInputQueue.SendCommand "CT= "

    End If  ' Color Table

    If DialogBoxName = "Color Table" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultCancel

    End If  ' Color Table

    If DialogBoxName = "Design File Settings" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Import Levels" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd Electrical.levels.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Import Levels

    If DialogBoxName = "Level/Filter Import" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Level/Filter Import

    If DialogBoxName = "Open" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultCancel

    End If  ' Open

    If DialogBoxName = "Open" Then
```

```
        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd P:\Active Projects\PGE\Substation\6446 BELL\2000 Su
bstation\2300 Engineering\2310 Electrical\2311 Drawings\Indoor\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd bell7313b0.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Open

    If DialogBoxName = "Design File Settings" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultCancel

    End If  ' Design File Settings

    If DialogBoxName = "Color Table" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultCancel

    End If  ' Color Table

    If DialogBoxName = "Design File Settings" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Design File Settings" Then

    '   The following statement opens modal dialog "Advanced Unit Settings"

    '   The following statement opens modal dialog "Alert"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Alert" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Alert

    If DialogBoxName = "Advanced Unit Settings" Then

        SetCExpressionValue "dgnSet.adv_uorPerStorage", 10000, "DGNSET"
    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK
    End If  ' Advanced Unit Settings

    If DialogBoxName = "Alert" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Alert

    If DialogBoxName = "Color Table" Then

        CadInputQueue.SendCommand "CT= "

    End If  ' Color Table

    If DialogBoxName = "Design File Settings" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Color Table" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultCancel

    End If  ' Color Table

    If DialogBoxName = "Import Levels" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd Electrical.levels.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Import Levels

    If DialogBoxName = "Level/Filter Import" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Level/Filter Import

    If DialogBoxName = "Compress Options" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Compress Options
    If DialogBoxName = "Import Levels" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd Electrical.levels.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Import Levels

    If DialogBoxName = "Level/Filter Import" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Level/Filter Import

    If DialogBoxName = "Open" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd P:\Active Projects\PGE\Substation\6446 BELL\2000 Su
bstation\2300 Engineering\2310 Electrical\2311 Drawings\Indoor\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd bell7313c0.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Open

    If DialogBoxName = "Color Table" Then

        CadInputQueue.SendCommand "CT= "

    End If  ' Color Table

    If DialogBoxName = "Color Table" Then

    End If  ' Color Table

    If DialogBoxName = "Import Levels" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd Electrical.levels.dgn"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Import Levels

    If DialogBoxName = "Level/Filter Import" Then
    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Level/Filter Import

    If DialogBoxName = "Design File Settings" Then

    '   The following statement opens modal dialog "Advanced Unit Settings"

    '   The following statement opens modal dialog "Alert"

    '   This only modifies a few bits of the variable it changes. It first
    '   creates a mask for clearing the bits it will change. Then it gets
    '   the variable and uses the mask to clear those bits. Finally
    '   it sets the desired bits in the value and saves the updated value.
        lngTemp = Not 3
        lngTemp = GetCExpressionValue("dgnSet.unitFormatDGN", "DGNSET") And lngTemp
        SetCExpressionValue "dgnSet.unitFormatDGN", lngTemp Or 1, "DGNSET"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Alert" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Alert

    If DialogBoxName = "Advanced Unit Settings" Then

        SetCExpressionValue "dgnSet.adv_uorPerStorage", 10000, "DGNSET"
    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Advanced Unit Settings

    If DialogBoxName = "Design File Settings" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Design File Settings" Then

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Design File Settings

    If DialogBoxName = "Color Table" Then
    End If  ' Color Table

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGESCH.CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

    If DialogBoxName = "Attach Cell Library" Then

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setDirectoryCmd J:\PGE\Cad Standards\PGE Cell Libraries\"

        CadInputQueue.SendCommand "MDL COMMAND MGDSHOOK,fileList_setFileNameCmd PGESHEET(new).CEL"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Attach Cell Library

End Sub
```

Macro6ModalHandler - 1

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Print Attributes" Then

    '   Set a variable associated with a dialog box
        SetCExpressionValue "plotAttrUI.line_wghts", 0, "PLOTDLG"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Print Attributes

End Sub
```

```
Implements IModalDialogEvents
Private Sub IModalDialogEvents_OnDialogClosed(ByVal DialogBoxName As String, ByVal DialogResult As MsdDialogBoxResult)

End Sub

Private Sub IModalDialogEvents_OnDialogOpened(ByVal DialogBoxName As String, DialogResult As MsdDialogBoxResult)

    If DialogBoxName = "Print Attributes" Then

    '   Set a variable associated with a dialog box
        SetCExpressionValue "plotAttrUI.line_wghts", 0, "PLOTDLG"

    '   Remove the following line to let the user close the dialog box.
        DialogResult = msdDialogBoxResultOK

    End If  ' Print Attributes

End Sub
```