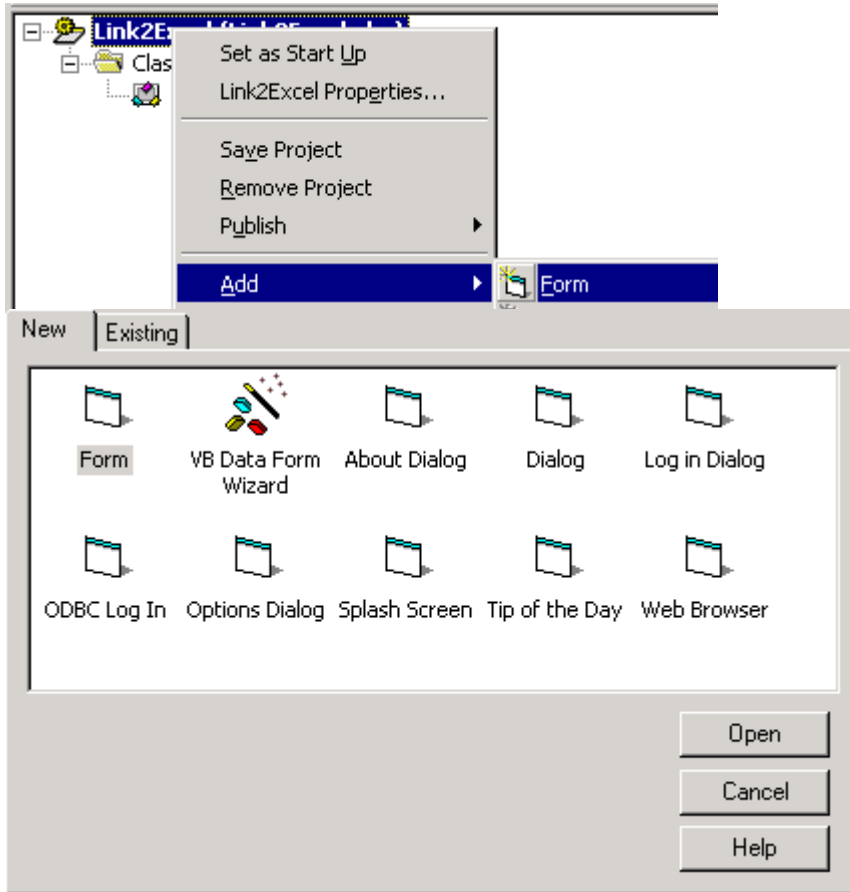
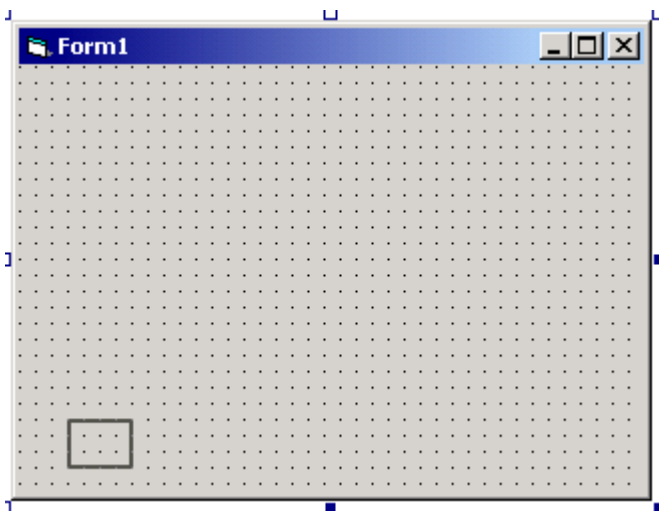



This subroutine is used to Place Fixed WorkPoints into a Part Document.

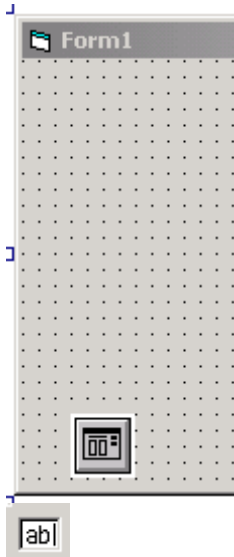


Highlight the project name.
Right click.
Select Add→Form.

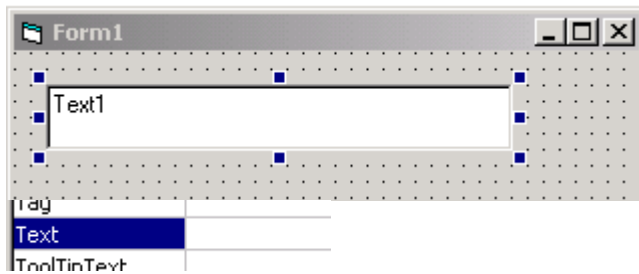
Select 'Form' and 'Open'.



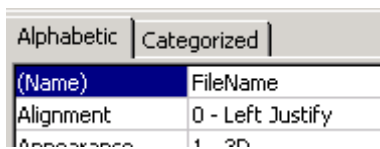
 Add the Common Dialog control to your program, depress the Control in your toolbar and then draw a small rectangle on your form.



The control icon will appear in the rectangle.
The icon will be invisible when your program is running.



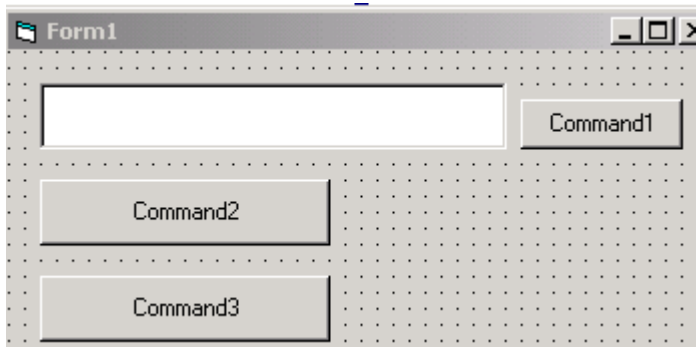
Select the Text box control and place on the form.
Select the text box control.
Right click and select 'Properties'.



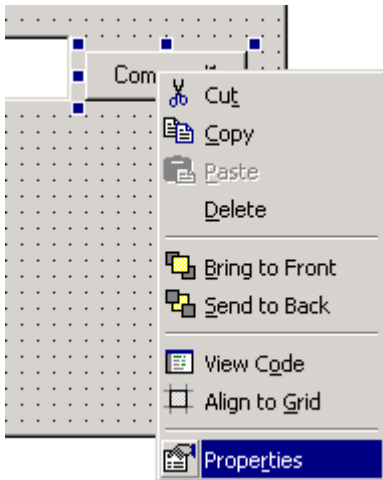
Erase any default text in the Text field.
Change the Name to FileName.



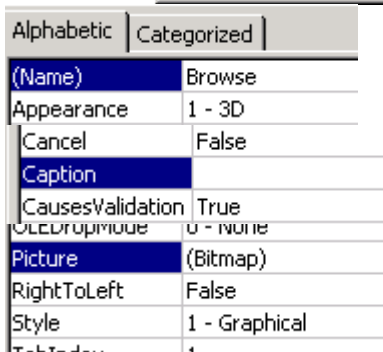
Close the properties dialog.
Select the Command Button control.
Drag the command button three times onto the form.



Place the Command Buttons as shown.



Select the first command button.
Right click and select 'Properties'.



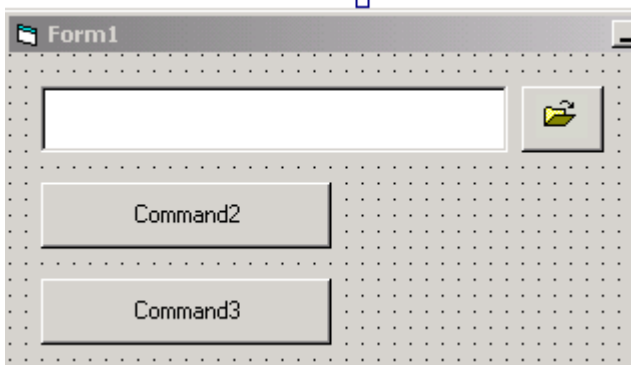
This button will show a picture or icon.
You can apply bitmaps or jpeg files to your buttons.
Change the name of the button to Browse.

Erase any text in the Caption field.

Select the ... button in the Picture field
and select a bmp file.

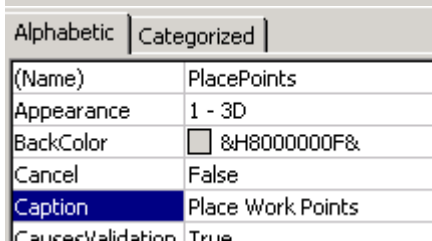
You can download a file called
'open.bmp' file from the publisher's
website.

Set the Style to Graphical.

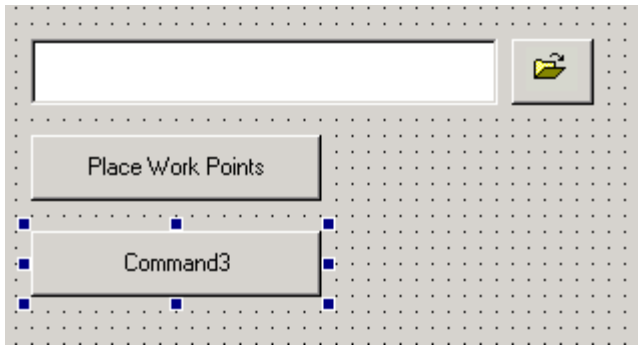


Close the Properties
dialog.

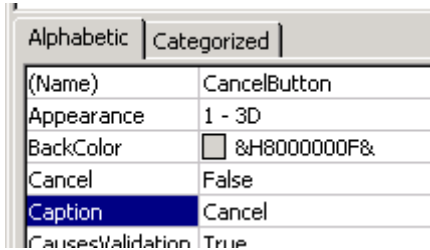
We see the
open.bmp file in the
command button.



Set the Properties
for the Second
Command Button.
Change the Name
to PlacePoints.
Change the Caption
to Place Work
Points.



Select the
Command Button 3.
Right click and
select 'Properties'.



Change the Name to Cancel.
Change the Caption to 'Cancel'.

Double click on the button with the browse/open picture to switch to the code window.
Enter the code shown.

```
Private Sub Browse_Click()  
    'filter for excel files  
    CommonDialog1.Filter = "Excel Files (*.xls)|*.xls|All Files (*.*)|*.*"  
    CommonDialog1.FilterIndex = 1  
    'open file dialog  
    CommonDialog1.ShowOpen  
    'store selected file name with path  
    WhatFile = CommonDialog1.FileName  
    'Display file name in text box  
    Form1.Text1.Text = WhatFile  
End Sub
```

- 1) We define the filter to applied to the open file dialog box.
In this case, we only want the dialog to list Excel files.
- 2) We enable the filter by setting it to 1.
- 3) We show the Open dialog.
- 4) We store the selected filename in the WhatFile variable.
- 5) We display the selected file name in the text box.

You can test to see how this works by pressing the Run button.



We see that the text box displays
the path and file name.
We can extend the text box so it
displays the entire path and file

name or change the display to
show only the file name.

```
CommonDialog1.ShowOpen  
'store selected file name with Path  
WhatFile = CommonDialog1.FileName  
'store selected file name  
FileName = CommonDialog1.FileTitle  
'Display file name in text box  
Form1.Text1.Text = FileName
```

To display only the file name without the path, change the code as shown using
CommonDialog1.FileTitle instead of CommonDialog1.FileName

Run the program to see the difference.

Double click on the Cancel button.

```
Private Sub Cancel_Click()  
End  
End Sub
```

Enter the code shown.
This causes the routine to exit.

Double click on the Place Points button.

```
'Excel Variables  
Dim oXL As Object  
Dim iMaxCol As Integer  
Dim iMaxRow As Integer  
Dim oXLSheet As Object  
Dim oXLUsed As Object
```

We start by defining the Excel variables.
These are the variables used to call and
manage the Excel spreadsheet.

```
'Point Variables  
Dim XPoint As Double  
Dim YPoint As Double  
Dim ZPoint As Double
```

Next we define the point variables used
to locate the fixed work points.

```
'Inventor variables  
Dim IV As Object  
Dim oCompDef As PartComponentDefinition  
Dim oTG As TransientGeometry  
Dim oWPs As WorkPoint
```

Then, we define the variables used by Inventor.

```
On Error Resume Next
  Set oXL = GetObject(
"Excel.Application")
  If Err <> 0 Then
    Set oXL =
CreateObject("Excel.Application")
    oXL.Visible = False
    oXL.UserControl = True
  Exit Sub
End If

On Error GoTo 0
```

The next section of code checks if Excel is running.

If Excel is not running, then the else statement opens a session of Excel. The oXL.Visible = False sets Excel to invisible.

```
oXL.Workbooks.Open WhatFile
```

The next line of code opens the file the user selected.

```
'The entire used area of the sheet CellA1 to the lower rightmost cell containing any
'Data
Set oXLSheet = oXL.ActiveWorkbook.Sheets("sheet1")
Set oXLUsed = oXLSheet.UsedRange

'Highest column number of the "Used range"
iMaxCol = oXLUsed.Columns(oXLUsed.Columns.Count).Column

'Highest row number of the "Used range"
iMaxRow = oXLUsed.Rows(oXLUsed.Rows.Count).Row
```

The next section of code checks to see which columns and rows to use to extract data from.

Since we are assuming just three columns are used (for the X,Y, and Z coordinates), we don't really need to worry about the iMaxCol values.

```
'Get the Inventor Application
Set IV = GetObject( "Inventor.Application")
' This assumes that it is a part document.
Dim oPartDoc As PartDocument
Set oPartDoc = IV.ActiveDocument
' Set a reference to the component definition.
Set oCompDef = oPartDoc.ComponentDefinition
' Set a reference to the transient geometry object.
Set oTG = IV.TransientGeometry
```

```
'set Inventor visible  
IV.Visible = True
```

The next section of code activates the active part document that is currently open in Inventor.

```
Dim iColCount As Integer  
Dim iRowCount As Integer  
iRowCount = 2  
  
For iRowCount = 2 To iMaxRow  
    XPoint = oXLSheet.Cells(iRowCount, 1).Value  
    YPoint = oXLSheet.Cells(iRowCount, 2).Value  
    ZPoint = oXLSheet.Cells(iRowCount, 3).Value  
    Set oWPs = oCompDef.WorkPoints.AddFixed(oTG.CreatePoint(XPoint, YPoint,  
ZPoint))  
    If Err Then Err.Clear  
Next iRowCount
```

The next section code is where the action really occurs.

We set the iRowCount to start at 2 (and ignore the top row which contains our headers) and process through to the last row.

For each row, we get the XPoint value, the Ypoint value, and Zpoint value.

We then create the fixed workpoint.

```
' unload the form  
Unload Me
```

Finally we unload the form.

	A	B	C	D
X		Y	Z	
	0	0	0	
	10	0	95	
	20	6	160	

This is how your spreadsheet should be set up.