VBA SOURCE CODE BOOK

Excel For Freelancers

Learn How To Add Bar Codes & Pictures While Creating Unlimited Labels In Excel

DOWNLOAD APPLICATION

WATCH ENTIRE SERIES

VIEW TRAINING

by: Randy Austin

ABOUT THE AUTHOR

A two-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With nearly 300,000 YouTube subscribers, 14,000,000 video views, 200+ comprehensive training videos, and a thriving 40,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:

WEBSITE YOUTUBE FACEBOOK TWITTER

INSTAGRAM TELEGRAM RUMBLE

MVP Microsoft® Most Valuable Professional

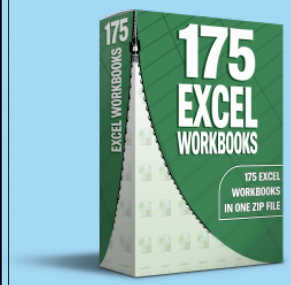
OUR COURSES & PRODUCTS

This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

Click here to learn more

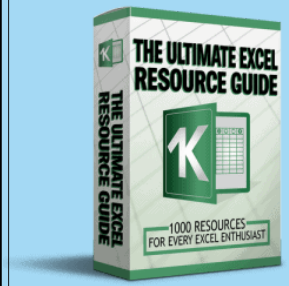
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

Click here to learn more



Incredible Package of 175 of my BEST Applications into a SINGLE ZIP File which also includes the "175 Workbook Library".

Click here to learn more



With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

Click here to learn more

```
Option Explicit

Private Sub Worksheet_Change(ByVal Target As Range)
  If Not Intersect(Target, Range("E8" )) Is Nothing Then
    Range("B4" ).Value = True 'Set Template Change to True
    LoadLabel 'Load Label details on change of label
  End If
  If Not Intersect(Target, Range("I21" )) Is Nothing Then
    If Cells(Target.Row, Target.Column).Value <> Empty Then GetDataDetails 'Get Destinationworksheet data on worksheet change
  End If

  'On Change of Column Format, when Column D is not empty, and not on Mapping load, then Update Format Validation and Value
  If Not Intersect(Target, Range("H27:H77" )) Is Nothing And Range("D"  & Target.Row).Value <> Empty Then
    Range("I"  & Target.Row).Validation.Delete
    If Range("H"  & Target.Row).Value <> "General"  Then Range("I"  & Target.Row).Validation.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween, Formula1:="="  & Range("H"  & Target.Row).Value & "FormatList"
  End If

  'On Change Of Format, update reformat sample Data
  If Not Intersect(Target, Range("I27:I77" )) Is Nothing And Range("I"  & Target.Row).Value <> Empty Then
    Range("E"  & Target.Row).NumberFormat = Range("I"  & Target.Row).Value  'Update Format for sample data
  End If
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
  If Target.CountLarge > 1 Then Exit Sub
  Dim DesignRow As Long, DesignColumn As Long
  Dim ShapeName As String, ShapeFormula As String, DataValue
  If Target.CountLarge > 5 Then Exit Sub
  'Display Filter Options on Selection
  If Not Intersect(Target, Range("G27:G77" )) Is Nothing And Range("D"  & Target.Row).Value <> Empty And Target.Count < 2 Then
    Range("B1" ).Value = Target.Row
    ShowFilterGroup 'Display Text Filter Buttons
  Else: 'Other than Filter Selection, hide filterGroup
    Shapes("FiltGrp" ).Visible = msoFalse
  End If

  'Check/Uncheck Data & Label Options
  If Not Intersect(Target, Range("F83:G132" )) Is Nothing And Range("D"  & Target.Row).Value <> Empty Then
    DesignColumn = Target.Column 'Selected Design Column
    DesignRow = Target.Row 'Selected Design Row
    If DesignColumn = 6 Then ShapeName = "Label"  & DesignRow Else ShapeName = "Data"  & DesignRow
    If DesignColumn = 6 Then ShapeFormula = "=$D$"  & DesignRow Else ShapeFormula = "=$E$"  & DesignRow
    If DesignColumn = 7 Then DataValue = Range("E"  & DesignRow).Value 'Get value of data to check for possible pictures
    If Target.Value = "ü"  Then
      Target.ClearContents
      On Error Resume Next
      Shapes(ShapeName).Delete 'Delete shape
      If DesignColumn = 7 Then Shapes("ItemPicture"  & DesignRow).Delete 'Delelete any possible picture
```

```
1 2 3
49          On Error GoTo 0
50        Else:
51          Target.Value = "ü"
52
53        If InStr(DataValue, "\" ) > 1 Then  'Some type of file (possible picture)
54          If InStr(DataValue, ".jpg" ) > 0 Or InStr(DataValue, ".jpeg" ) > 0 Or InStr(
            DataValue, ".png" ) > 0 Or InStr(DataValue, ".bmp" ) > 0 Or InStr(DataValue,
            ".tiff" ) > 0 Then 'Confirm Picture File
55            ShapeName = "ItemPicture"  & DesignRow 'Reset Shape Name To Picture
56          If Dir(DataValue, vbDirectory) = ""  Then
57            MsgBox "File Path for this picture is not value"
58            Exit Sub
59          End If
60
61          With Pictures.Insert(DataValue)
62            With .ShapeRange
63              .LockAspectRatio = msoTrue
64              .Height = 95
65              .Name = ShapeName
66            End With 'Shape Range
67            With Shapes(ShapeName)
68              .Left = Sheet1.Range("H84" ).Left
69              .Top = Sheet1.Range("H84" ).Top
70              .IncrementLeft 20
71              .IncrementTop 15
72              '.ZOrder msoSendToBack
73            End With
74             ' .Shapes("BackLabShape").ZOrder msoSendToBack
75          End With
76        End If 'End If Picture
77        Else: 'Text File (Not Picture)
78          'Create & Hide Label Shapes
79          Shapes("SampleTextBox" ).Copy
80          Range("H83" ).Select
81          Range("H83" ).Select
82          Sheet1.Paste
83          Selection.Name = ShapeName
84          With Shapes(ShapeName)
85            .Left = Sheet1.Range("H83" ).Left
86            .Top = Sheet1.Range("H83" ).Top
87            .IncrementLeft 15
88            .IncrementTop 10
89            .DrawingObject.Formula = ShapeFormula
90            .Placement = xlFreeFloating
91            .Visible = msoCTrue
92          End With
93        End If
94      End If
95      Range("E"  & Target.Row).Select 'Select out of the cell
96    End If
97  End Sub
98
```

```
1   Option Explicit
2
```

E

```
1   Option Explicit
2
```

E

```
1   Option Explicit
2
```

E

```
1    Option Explicit
2
```

E

```
Option Explicit
Dim StartTime As Double
Dim SecondsElapsed As Double

Sub CalculateRunTime_Minutes()
    'PURPOSE: Determine how many minutes it took for code to completely run
    'SOURCE: www.TheSpreadsheetGuru.com/the-code-vault

    Dim StartTime As Double
    Dim MinutesElapsed As String

    'Remember time when macro starts
    StartTime = Timer

    '****************************
    'Insert Your Code Here...
    '****************************

    'Determine how many seconds code took to run
    MinutesElapsed = Format((Timer - StartTime) / 86400, "hh:mm:ss" )

    'Notify user in seconds
    MsgBox "This code ran successfully in "  & MinutesElapsed & " minutes" , vbInformation

End Sub

Sub ResetCalc()
    With Application
        .EnableEvents = True
        .Calculation = xlCalculationAutomatic
        .ScreenUpdating = True
    End With
End Sub



Sub StartClock()

    'Remember time when macro starts
    StartTime = Timer
End Sub

Sub StopCalc()
    With Application
        .EnableEvents = False
        .Calculation = xlCalculationManual
        '.ScreenUpdating = False
    End With
End Sub

Sub StopClock()

    'Determine how many seconds code took to run
    SecondsElapsed = Round(Timer - StartTime, 2)

    'Notify user in seconds
    MsgBox "This code ran successfully in "  & SecondsElapsed & " seconds" , vbInformation
End Sub
```

```
Option Explicit
Sub BuildLabelSheet()
  'If Sheet1.Range("B4").Value = False Then Exit Sub 'No need to rebuild label sheet unless template has changed
  StopCalc
  Dim LabSheetQty As Long, LabelCol As Long, LabelRow As Long, SetCount As Long
  Dim LastLabelRow As Long, LastLabelCol As Long, SetCol As Long, SetRow As Long
  Dim DataRow As Long, LastDataRow, MaxLabels As Long, MaxRows As Long
  Dim LabHeight As Double, LabWidth As Double, ColumnSpace As Double, RowSpace As Double
  Dim LeftMargin As Double, RightMargin As Double, TopMargin As Double, BottomMargin  As Double
  Dim PrintArea As Range
  With Sheet1
    LabSheetQty = .Range("E9" ).Value 'Label Sheet Qty
    LabelCol = .Range("E11" ).Value 'Number of label columns
    LabelRow = .Range("G11" ).Value 'Number of label rows
    LabHeight = .Range("E10" ).Value 'Label Height
    LabWidth = .Range("G10" ).Value   'Label Width
    ColumnSpace = .Range("E13" ).Value  'Column Spacing
    RowSpace = .Range("G13" ).Value  'Row Spacing
    LeftMargin = .Range("E14" ).Value 'Left Margin Spacing
    RightMargin = .Range("G14" ).Value 'Right Margin
    TopMargin = .Range("E15" ).Value 'Top Margin
    BottomMargin = .Range("G15" ).Value 'Bottom Margin
    MaxLabels = .Range("H22" ).Value - 1 'Maximum # of Labels (without any filter)
    MaxRows = MaxLabels / LabelCol
    'Determine the Total # of Rows (Full Sheets) Round Up To Whole # (Maximum Rows / Rows Per sheet) * Number OF Rows Per Sheet
    LastLabelRow = Application.WorksheetFunction.RoundUp((MaxRows / LabelRow), 0) * LabelRow
    LastLabelCol = LabelCol + (LabelCol - 1)
    LastLabelRow = LastLabelRow + (LastLabelRow - 1) 'Set the Last Label Row (Use full Sheets)
  End With
  With Sheet3 'Label Template
    'Setup Columns
    For SetCol = 1 To LastLabelCol Step 2
      For SetCount = 1 To 3 'Run 3 times to set width
        .Cells(1, SetCol).ColumnWidth = Application.InchesToPoints(LabWidth) * (.Cells(1, SetCol).ColumnWidth / .Cells(1, SetCol).Width) - 3
      Next SetCount
      If SetCol <> LastLabelCol Then .Cells(1, SetCol + 1).ColumnWidth = ColumnSpace
    Next SetCol
    'Setup Rows
    For SetRow = 1 To LastLabelRow Step 2
      .Cells(SetRow, 1).RowHeight = Application.InchesToPoints(LabHeight) 'Set Row Height

      .Cells(SetRow + 1, 1).RowHeight = RowSpace
    Next SetRow
    With .PageSetup
      .PrintArea = Range(Sheet3.Cells(1, 1), Sheet3.Cells(LastLabelRow, LastLabelCol)).Address
      .LeftMargin = Application.InchesToPoints(LeftMargin)
      .RightMargin = Application.InchesToPoints(RightMargin)
      .TopMargin = Application.InchesToPoints(TopMargin)
      .BottomMargin = Application.InchesToPoints(BottomMargin)
      .HeaderMargin = 0
      .FooterMargin = 0
    End With
    Sheet1.Range("B4" ).Value = False 'Reset Label Change to False
  End With
1
```

```
    ResetCalc
End Sub
Sub FilterDataCreateLabels()
    Dim FirstFiltCol As Long, LastFiltCol As Long, TotalDataCol As Long
    Dim LabelCol As Long, LabelRow As Long, LabelNumber As Long
    Dim TotalLabelCol As Long, TotalLabelRow As Long
    Dim FirstLabelRow As Long, LastLabelRow As Long, LabelDataRow As Long
    Dim LastLabelCol As Long, LabelDataCol As Long
    Dim DesignRow As Long, LastDesignRow As Long
    Dim FirstDataRow As Long, LastDataRow As Long, LastHeaderRow As Long
    Dim LabelTemplate As Shape, TempLabelShape As Shape, LabelShape As Shape, ItemPic As Shape

    Dim LabelRng As Range, LabelCell As Range
    Dim PicFileName As String
    ''Remove any labels currently on the sheet
    For Each LabelShape In Sheet3.Shapes
      LabelShape.Delete
    Next
    Set LabelShape = Nothing
    'Remove any Label Preview (if existing)
    On Error Resume Next
    Sheet1.Shapes("LabPreview" ).Delete
    On Error GoTo 0

    With Sheet1
      TotalDataCol = .Range("J23" ).Value
      FirstFiltCol = TotalDataCol + 5
      LastFiltCol = FirstFiltCol + TotalDataCol
      Range(Sheet4.Cells(1, FirstFiltCol), Sheet4.Cells(1, LastFiltCol)).Value = Range(Sheet4
      .Cells(1, 1), Sheet4.Cells(1, TotalDataCol + 1)).Value 'Copy Over for headers criteria

      Range(Sheet4.Cells(3, FirstFiltCol), Sheet4.Cells(3, LastFiltCol)).Value = Range(Sheet4
      .Cells(1, 1), Sheet4.Cells(1, TotalDataCol + 1)).Value 'Copy Over headers for results
      LastHeaderRow = Application.WorksheetFunction.CountA(.Range("D27:D77" )) + 26 'Last
      header Row
      If LastHeaderRow < 27 Then Exit Sub 'On No Headers
      .Range("G27:G"  & LastHeaderRow).Copy
      Sheet4.Cells(2, FirstFiltCol).PasteSpecial xlPasteValues, , , True 'Copy Over filters
      LastDataRow = .Range("H22" ).Value - 1 'Last Row Of Data
      Range(Sheet4.Cells(1, 1), Sheet4.Cells(LastDataRow, TotalDataCol + 1)).AdvancedFilter
      xlFilterCopy, _
      CriteriaRange:=Range(Sheet4.Cells(1, FirstFiltCol), Sheet4.Cells(2, LastFiltCol)),
      CopyToRange:=Range(Sheet4.Cells(3, FirstFiltCol), Sheet4.Cells(3, LastFiltCol)), Unique
      :=True

      'Builld Labels On Label Page
      On Error Resume Next
      Set LabelTemplate = Sheet1.Shapes("TemplateGroup" )
      On Error GoTo 0
      If LabelTemplate Is Nothing Then
        GroupTemplateShapes 'Run Macro to Create Label Group
        On Error Resume Next
        Set LabelTemplate = Sheet1.Shapes("TemplateGroup" ) '2nd ry
        On Error GoTo 0
      End If
      If LabelTemplate Is Nothing Then Exit Sub 'Exit on 2nd Failure to set locate label
      template
      LastDataRow = Sheet4.Cells(99999, FirstFiltCol).End(xlUp).Row 'Last Results Row
      .Range("E141" ).Value = LastDataRow - 3 'Display # Of labels to Print
      .Range("G141" ).Value = Application.WorksheetFunction.RoundUp((LastDataRow - 3) / .
      Range("E9" ).Value, 0)  'Total # Of Labels Divided by labels per sheet
```

```
105         TotalLabelCol = .Range("E11" ).Value 'Number of label columns
106         TotalLabelRow = Application.WorksheetFunction.RoundUp((LastDataRow - 3) / TotalLabelCol
            , 0) 'Number of label rows
107         LastLabelCol = TotalLabelCol + (TotalLabelCol - 1)
108         LastLabelRow = TotalLabelRow + (TotalLabelRow - 1)
109         LastDesignRow = LastHeaderRow + 56
110         Sheet3.Activate
111         LabelRow = 1
112         LabelCol = 1
113         LabelNumber = 1
114         LabelTemplate.Copy
115         Sheet3.Activate
116
117         Sheet3.Range("A1" ).Select 'Paste in Temporary shape
118         Sheet3.Paste Destination:=Sheet3.Range("A1" )
119         Set TempLabelShape = ThisWorkbook.Sheets("Printed Labels" ).Shapes("TemplateGroup" )
120
121         LabelDataRow = 4 'Set First Row of Data (Data sheet, filtered row)
122
123       For LabelRow = 1 To LastLabelRow Step 2 'First Loop Through Rows
124         For LabelCol = 1 To LastLabelCol Step 2 'Second Loop Through Columns
125
126           Set LabelCell = ThisWorkbook.Sheets("Printed Labels" ).Cells(LabelRow, LabelCol)
127           TempLabelShape.Copy 'Copy Temporary Label Shape
128           LabelCell.Select 'Select Cell To Place shape in
129           On Error GoTo ErrorHandler
130           Sheet3.Paste Destination:=LabelCell 'Place Shape in Label cell
131
132         With TempLabelShape 'WIth temp. label shape
133             .Left = Sheet3.Cells(LabelRow, LabelCol).Left
134             .Top = Sheet3.Cells(LabelRow, LabelCol).Top
135             .Visible = msoCTrue
136             .Name = "Label"  & LabelNumber 'Rename shape group, specific to label #
137         End With
138
139           LabelDataCol = FirstFiltCol 'Set the first column of filtered data to move
              through
140         For DesignRow = 83 To LastDesignRow 'Loop Through Design Rows to see which ones
            are selected to be visible in the label
141
142           If .Range("G"  & DesignRow).Value = "ü"  Then 'Show Data (or picture)
143
144             'Check For Item Picture
145           If InStr(Sheet4.Cells(LabelDataRow, LabelDataCol).Value, "\" ) > 1 Then
              'File found (possible picture)
146
147             On Error Resume Next
148             Set ItemPic = Sheet1.Shapes("ItemPicture"  & DesignRow) 'check for
                correct file name and set picture
149             On Error GoTo 0
150           If Not ItemPic Is Nothing Then 'Picture exists if its found
151             With Sheet3.Pictures.Insert(Sheet4.Cells(LabelDataRow, LabelDataCol).
                Value)
152               With .ShapeRange
153                 .Name = "ItemLabelPicture"  & DesignRow & "_"  & LabelNumber
                    'Assign Unique name to picture based on Design Row and Label #
154               End With
155             End With
156             With Sheet3.Shapes("ItemLabelPicture"  & DesignRow & "_"  &
                LabelNumber)
```

```
                .Height = ItemPic.Height 'Set Height same as original picture
                .Width = ItemPic.Width 'Set width same as original pciture
                .Left = Sheet3.Cells(LabelRow, LabelCol).Left 'Set intial Left position based on cell
                .Top = Sheet3.Cells(LabelRow, LabelCol).Top 'Set intial Top Position based on cell
                .IncrementLeft ItemPic.Left - Sheet1.Shapes("BackLabShape" ).Left 'Increment left (move to right) based on distance from Label backround and picture
                .IncrementTop ItemPic.Top - Sheet1.Shapes("BackLabShape" ).Top 'Increment top (move down) based on distance from Label backround and picture
                .Visible = msoCTrue 'Display Picture
            End With

        End If

    Else ' Not Picture
      'On Error Resume Next

      Sheet3.Shapes("Data"  & DesignRow).DrawingObject.Formula = "='Imported Data'!"  & Sheet4.Cells(LabelDataRow, LabelDataCol).Address 'Get Data Value
      Sheet3.Shapes("Data"  & DesignRow).Name = "Label"  & LabelNumber & "_Data"  & DesignRow
      'On Error GoTo 0
      Sheet1.Shapes("Data"  & DesignRow).PickUp
      Sheet3.Shapes("Label"  & LabelNumber & "_Data"  & DesignRow).Apply
    End If
  End If
  On Error Resume Next
  Sheet3.Shapes("ItemPicture"  & DesignRow).Delete 'Remove Template picture copied over
  On Error GoTo 0
  LabelDataCol = LabelDataCol + 1 'Increment Data Column

 Next DesignRow


  Set TempLabelShape = Sheet3.Shapes("TemplateGroup" ) 'Reset Temp Label To Original sheet Template

  LabelNumber = LabelNumber + 1
  LabelDataRow = LabelDataRow + 1
  If LabelDataRow > LastDataRow Then Exit For
 Next LabelCol
 If LabelDataRow > LastDataRow Then Exit For
Next LabelRow


On Error Resume Next
Sheet3.Shapes("TemplateGroup" ).Delete 'Remove temporary group
On Error GoTo 0
If LabelRow > 10 Then LabelRow = 10 'Set max height for sample picture
Set LabelRng = Range(Sheet3.Cells(1, 1), Sheet3.Cells(LabelRow, LastLabelCol))
Application.CutCopyMode = False
Application.Wait Now + 0.00001
LabelRng.CopyPicture
Sheet1.Activate
Application.Wait Now + 0.00001
Sheet1.Range("D143" ).PasteSpecial
```

```
1 2
207         Selection.Formula = "'Printed Labels'!"  & LabelRng.Address
208         Selection.Name = "LabPreview"
209         Application.CutCopyMode = False
210         With .Shapes("LabPreview" )
211           .IncrementLeft 20
212         End With
213         Sheet1.Range("F144" ).Select
214
215       End With
216       Exit Sub
217       ErrorHandler:
218       Application.Wait Now + 0.00001
219       TempLabelShape.Copy
220       Application.Wait Now + 0.00001
221       Sheet3.Paste Destination:=LabelCell
222       Resume Next
223     End Sub
224
225     Sub PrintLabels()
226       Sheet3.PrintOut , , , False, True, , , , False
227     End Sub
228
229
230
```

```
Option Explicit

Sub BrowseForData()
  Dim FileFldr As FileDialog
  Dim DataFilePath As String
  Dim DataWs As Worksheet
  Dim ThisWkBk As Workbook, DataWkBk As Workbook
  Dim WsRow As Long

  Set ThisWkBk = ThisWorkbook
  With Sheet1
    WsRow = 20
    .Range("A20:A50,E20:H20,I21:J21,F22:F23,H22:H23" ).ClearContents 'Clear Any older data details
    .Range("D27:J77" ).ClearContents 'Clear any older Worksheet data
    Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
    With FileFldr
      .Title = "Select Data File"
      .Filters.Add "Excel Files" , "*.xlsm,*.xlsx,*.xlsb,*.xls"
      If .Show <> -1 Then GoTo NoSelection
      Sheet1.Range("E20" ).Value = .SelectedItems(1) 'Get Full File Path
    End With

    DataFilePath = .Range("E20" ).Value 'Long File Path
    .Range("E21" ).Value = Dir(DataFilePath, vbDirectory) 'File Name
    Workbooks.Open (DataFilePath) 'Open Data Workbook
    Set DataWkBk = Workbooks(Dir(DataFilePath)) 'Set Data Workbook

    For Each DataWs In DataWkBk.Worksheets
      Sheet1.Range("A"  & WsRow).Value = DataWs.Name
      WsRow = WsRow + 1
    Next
    ThisWkBk.Activate
    .Range("I21" ).Select 'Select cell for user to select sheet
    NoSelection:
  End With
End Sub


Sub GetDataDetails()
  Dim DataFilePath As String, DataShtName As String
  Dim WsRow As Long, FirstRow As Long, TableRow As Long, TableCol As Long, FirstCol As Long, LastRow As Long, LastCol As Long
  Dim DataRow As Long, DataCol As Long, DesignRow As Long
  Dim DataWkBk As Workbook, ThisWkBk As Workbook

  Set ThisWkBk = ThisWorkbook
  With Sheet1
    DataFilePath = .Range("E20" ).Value 'Data Workbook File Path
    On Error GoTo MissingWorkbook
    Set DataWkBk = Workbooks(Dir(DataFilePath)) 'Set Data Workbook
    On Error GoTo 0
    DataShtName = .Range("I21" ).Value 'Data Worksheet Name
    .Range("D27:J77" ).ClearContents 'Clear any older Worksheet data
    .Range("D83:G132" ).ClearContents 'Clear any  Lables & Data for design steps
  End With


  With DataWkBk.Sheets(DataShtName)

    'Get Last Row Of Data
```

```
60          'On Error Resume Next
61          LastRow = .Cells.Find(What:="*" , After:=.Range("A1" ), _
62          Lookat:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByRows, SearchDirection:=xlPrevious,
             MatchCase:=False).Row
63          'Get Last Column Of Data
64          LastCol = .Cells.Find(What:="*" , After:=.Range("A1" ), _
65          Lookat:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByColumns, SearchDirection:=
            xlPrevious, MatchCase:=False).Column
66          On Error GoTo 0
67
68          'Determine First Row Of Table w/ 70% of the cells filled
69          For TableRow = 1 To LastRow
70            If (Application.WorksheetFunction.CountA(Range(.Cells(TableRow, 1), .Cells(TableRow,
               LastCol))) / LastCol) >= 0.7 Then
71              FirstRow = TableRow
72              Exit For
73            End If
74          Next TableRow
75
76          'Determine First Column Of Table w/ 50% of the cells filled
77          For TableCol = 1 To LastCol
78            If (Application.WorksheetFunction.CountA(Range(.Cells(1, TableCol), .Cells(LastRow,
              TableCol))) / LastRow) >= 0.5 Then
79              FirstCol = TableCol
80              Exit For
81            End If
82          Next TableCol
83
84        End With
85
86        If Sheet1.Range("B18" ).Value = True Then FirstRow = FirstRow + 1 'Has Headers
87
88        Sheet1.Range("F22" ).Value = FirstRow 'Set First Row Of Data Data
89        Sheet1.Range("H22" ).Value = LastRow 'Set Last Row Of Data Data
90        Sheet1.Range("F23" ).Value = FirstCol 'Set First Column Of Data Data
91        Sheet1.Range("H23" ).Value = LastCol 'Set Last Column Of Data Data Data
92
93        ThisWkBk.Activate
94        'Load Data Details
95
96        DataRow = 27
97        DesignRow = 83
98        With Sheet1
99          If .Range("B18" ).Value = True Then 'Data has Headers
100
101           For DataCol = FirstCol To LastCol
102             .Range("D"  & DataRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow - 1,
                DataCol).Value 'Load Headers
103             .Range("D"  & DesignRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow - 1,
                 DataCol).Value 'Load Headers
104             DataRow = DataRow + 1
105             DesignRow = DesignRow + 1
106           Next DataCol
107         End If
108         'Add First Row Of Data
109         DataRow = 27
110         DesignRow = 83
111         For DataCol = FirstCol To LastCol
112           .Range("E"  & DataRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow, DataCol)
              .Value
113           .Range("E"  & DesignRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow,
```

```
                DataCol).Value  'Load First Row OF Data
114             DataRow = DataRow + 1
115             DesignRow = DesignRow + 1
116           Next DataCol
117           On Error Resume Next
118           Sheet1.Shapes("TemplateGroup" ).Delete 'Remove all data & label fields from any
              previuos label
119           On Error GoTo 0
120         End With
121         Exit Sub
122         MissingWorkbook:
123         MsgBox "Please check for a correct Data Workbook"
124       End Sub
125
126
127
```

```
Option Explicit

Sub AddFilt_Numb_GreaterThan()
  Dim DataRow As Long
  Dim FilterBy As String
  With Sheet1
    If .Range("B1" ).Value = Empty Then Exit Sub
    DataRow = .Range("B1" ).Value
    If .Range("G"  & DataRow).Value = Empty Then Exit Sub
    FilterBy = .Range("G"  & DataRow).Value 'Filter By Text
    FilterBy = Replace(Replace(Replace(FilterBy, "*" , "" ), "<" , "" ), ">" , "" )
    'Remove any previous filtering characters
    .Range("G"  & DataRow).Value = ">"  & FilterBy
  End With
End Sub


Sub AddFilt_Numb_LessThan()
  Dim DataRow As Long
  Dim FilterBy As String
  With Sheet1
    If .Range("B1" ).Value = Empty Then Exit Sub
    DataRow = .Range("B1" ).Value
    If .Range("G"  & DataRow).Value = Empty Then Exit Sub
    FilterBy = .Range("G"  & DataRow).Value 'Filter By Text
    FilterBy = Replace(Replace(Replace(FilterBy, "*" , "" ), "<" , "" ), ">" , "" )
    'Remove any previous filtering characters
    .Range("G"  & DataRow).Value = "<"  & FilterBy
  End With
End Sub

Sub AddFilt_Text_Contains()
  Dim DataRow As Long
  Dim FilterBy As String
  With Sheet1
    If .Range("B1" ).Value = Empty Then Exit Sub
    DataRow = .Range("B1" ).Value
    If .Range("G"  & DataRow).Value = Empty Then Exit Sub
    FilterBy = .Range("G"  & DataRow).Value 'Filter By Text
    FilterBy = Replace(Replace(Replace(FilterBy, "*" , "" ), "<" , "" ), ">" , "" )
    'Remove any previous filtering characters
    .Range("G"  & DataRow).Value = "*"  & FilterBy & "*"
  End With
End Sub

Sub AddFilt_Text_EndsWith()
  Dim DataRow As Long
  Dim FilterBy As String
  With Sheet1
    If .Range("B1" ).Value = Empty Then Exit Sub
    DataRow = .Range("B1" ).Value
    If .Range("G"  & DataRow).Value = Empty Then Exit Sub
    FilterBy = .Range("G"  & DataRow).Value 'Filter By Text
    FilterBy = Replace(Replace(Replace(FilterBy, "*" , "" ), "<" , "" ), ">" , "" )
    'Remove any previous filtering characters
    .Range("G"  & DataRow).Value = "*"  & FilterBy
  End With
End Sub
Sub AddFilt_Text_StartsWith()
  Dim DataRow As Long
  Dim FilterBy As String
1
```

```
1
58      With Sheet1
59        If .Range("B1" ).Value = Empty Then Exit Sub
60        DataRow = .Range("B1" ).Value
61        If .Range("G"  & DataRow).Value = Empty Then Exit Sub
62        FilterBy = .Range("G"  & DataRow).Value 'Filter By Text
63        FilterBy = Replace(Replace(Replace(FilterBy, "*" , "" ), "<" , "" ), ">" , "" )
          'Remove any previous filtering characters
64        .Range("G"  & DataRow).Value = FilterBy & "*"
65      End With
66    End Sub
67
68    Sub ShowFilterGroup()
69      Dim DataRow As Long
70      With Sheet1
71        If .Range("B1" ).Value = Empty Then Exit Sub
72        DataRow = .Range("B1" ).Value 'Data Row
73        With .Shapes("FiltGrp" )
74          .Left = Sheet1.Range("H"  & DataRow).Left
75          .Top = Sheet1.Range("H"  & DataRow).Top
76          .IncrementTop -15
77          .Visible = msoCTrue
78        End With
79      End With
80    End Sub
```

```
Option Explicit

Sub LoadLabel()
  With Sheet1
    Dim LabelRow As Long
    Dim PicName As String
    If .Range("B5" ).Value = Empty Then
      MsgBox "Please select a correct label from the drop down list"
      Exit Sub
    End If
    LabelRow = .Range("B5" ).Value 'Label Row
    .Range("E9" ).Value = Sheet2.Range("B"  & LabelRow).Value '# Labels Per Sheet
    .Range("G9" ).Value = Sheet2.Range("C"  & LabelRow).Value 'Type
    .Range("E10" ).Value = Sheet2.Range("D"  & LabelRow).Value 'Height
    .Range("G10" ).Value = Sheet2.Range("E"  & LabelRow).Value 'Width
    .Range("E11" ).Value = Sheet2.Range("F"  & LabelRow).Value 'Columns
    .Range("G11" ).Value = Sheet2.Range("G"  & LabelRow).Value 'Rows
    .Range("E13" ).Value = Sheet2.Range("I"  & LabelRow).Value 'Column Spacing
    .Range("G13" ).Value = Sheet2.Range("J"  & LabelRow).Value 'Row Spacing
    .Range("E14" ).Value = Sheet2.Range("K"  & LabelRow).Value 'Left Margin
    .Range("G14" ).Value = Sheet2.Range("L"  & LabelRow).Value 'Right Margin
    .Range("E15" ).Value = Sheet2.Range("M"  & LabelRow).Value 'Top Margin
    .Range("G15" ).Value = Sheet2.Range("N"  & LabelRow).Value 'Bottom Margin
    PicName = Sheet2.Range("H"  & LabelRow).Value 'Filename
    Sheet2.Shapes(PicName).Copy
    .Range("H8" ).Select
    .Paste
    On Error Resume Next
    .Shapes("LabSheetPic" ).Delete
    On Error GoTo 0
    With .Shapes(PicName)
      .Name = "LabSheetPic"
      .Left = Sheet1.Range("H8" ).Left
      .Top = Sheet1.Range("H8" ).Top
      .LockAspectRatio = msoCTrue
      .Width = 90
      .IncrementLeft 40
    End With
    .Range("E8" ).Select
  End With
End Sub
```

```
Option Explicit

Sub ImportLabelData()
  Dim DataFilePath As String, DataShtName As String
  Dim DataWkBk As Workbook, ThisWkBk As Workbook

  Dim FirstDataRow As Long, FirstDataCol As Long, LastDataCol As Long
  Dim LastDataRow As Long, DataCol As Long, DataRow As Long

  Dim TotalDataRows As Long, TotalDataColumns As Long
  Dim TableRow As Long
  Dim TableCol As Long
  Dim ColCount As Long
  Dim FirstAvailRow As Long


  Dim MapCol As Long
  Dim LastMapCol As Long
  Dim CellFormat As String

  With Sheet1
    'Make sure an proper file path
    DataFilePath = .Range("E20" ).Value 'Date File Path
    If Dir(DataFilePath) = ""  Then
      MsgBox "Please select a proper data file for your labels"
      CreateLabelStep2
      Exit Sub
    End If

    Set ThisWkBk = ThisWorkbook 'Set Current Workbook To Import

    'Get Data Workbook Variables
    On Error Resume Next 'An error will occur if we set a workbook that is not open. If the error occurs we can run a check and open the workbook
    Set DataWkBk = Workbooks(Dir(DataFilePath)) 'Set Data Workbook
    If DataWkBk Is Nothing Then 'Data Workbook is not open
      Set DataWkBk = Workbooks.Open(DataFilePath)
    End If
    DataShtName = .Range("I21" ).Value 'Data Worksheet Name
    FirstDataRow = .Range("F22" ).Value 'First Row of Data (excludes headers)
    FirstDataCol = .Range("F23" ).Value 'First Column Of Data from the Data sheet
    LastDataCol = .Range("H23" ).Value 'Last Data Column
    LastDataRow = .Range("H22" ).Value 'Last Row Of Data
    TotalDataRows = .Range("J22" ).Value 'Total Rows To Be  copied
    TotalDataColumns = .Range("J23" ).Value 'Total Columns To be copied
    Sheet4.Range("A1:ZZ9999" ).ClearContents 'Clear Any Existing Data

    'Copy Over headers (if any)
    If .Range("B18" ).Value = True Then Range(Sheet4.Cells(1, 1), Sheet4.Cells(1, TotalDataColumns)).Value = _
    Range(DataWkBk.Sheets(DataShtName).Cells(FirstDataRow - 1, FirstDataCol), DataWkBk.Sheets(DataShtName).Cells(FirstDataRow - 1, LastDataCol)).Value

    'Copy Over Data
    Range(Sheet4.Cells(2, 1), Sheet4.Cells(TotalDataRows + 1, TotalDataColumns)).Value = _
    Range(DataWkBk.Sheets(DataShtName).Cells(FirstDataRow, FirstDataCol), DataWkBk.Sheets(DataShtName).Cells(LastDataRow, LastDataCol)).Value
    ThisWorkbook.Activate
    Exit Sub
    DuplicateSheet:
    MsgBox "The Sheet Name '"  & .Range("I30" ).Value & "' already exists"  & vbCrLf &
1 2
```

```
1 2
        "Please select a new worksheet name to add"
58      End With
59  End Sub
60
61
62
63
```

```
Option Explicit

Sub Add_LabelPic()
  Dim PicFile As FileDialog
  With Sheet1
    Set PicFile = Application.FileDialog(msoFileDialogFilePicker)
    With PicFile
      .Title = "Select an Employee Picture"
      .Filters.Add "All Picture Files" , "*.jpg,*jpeg,*.gif,*.png,*.gif,*bmp,*.tiff" , 1
      If .Show <> -1 Then GoTo NoSelection
      Sheet1.Range("B84" ).Value = .SelectedItems(1) 'Put File Name in B84
    End With
    'Run Macro to Show Picture
    Show_LabelPic
    NoSelection:
  End With
End Sub

Sub GroupTemplateShapes()
  With Sheet1
    Dim LabelShape As Shape, TemplateGroup As Shape
    Dim ShapesNames As String
    Dim GrpArr As Variant
    For Each LabelShape In Sheet1.Shapes
      If InStr(LabelShape.Name, "Data" ) > 0 Or InStr(LabelShape.Name, "Label" ) > 0 Or
      InStr(LabelShape.Name, "LabelPic" ) Or InStr(LabelShape.Name, "ItemPicture" ) > 0
      Then
        ShapesNames = ShapesNames & LabelShape.Name & ","
      End If
    Next LabelShape
    If ShapesNames = Empty Then Exit Sub 'Grouped already or no shapes exist
    ShapesNames = Left(ShapesNames, Len(ShapesNames) - 1)
    GrpArr = Split(ShapesNames, "," )
    On Error GoTo NoGroup
    Set TemplateGroup = ThisWorkbook.Sheets("Label Creator" ).Shapes.Range((GrpArr)).Group
    With TemplateGroup
      .Name = "TemplateGroup"
      .LockAspectRatio = msoCTrue
      .Placement = xlMove
      .Visible = msoFalse
    End With
  End With
  NoGroup:
End Sub


Sub Show_LabelPic()

  Dim PicPath As String
  With Sheet1
    On Error Resume Next
    .Shapes("LabelPic" ).Delete 'Delete Picture if it exists
    On Error GoTo 0
    PicPath = .Range("B84" ).Value 'Path of the picture
    If PicPath = Empty Then Exit Sub
    With .Pictures.Insert(PicPath)
      With .ShapeRange
        .LockAspectRatio = msoTrue
        .Height = 95
        .Name = "LabelPic"
      End With 'Shape Range
```

```
    End With 'Pictures

    With .Shapes("LabelPic" )
      .Left = Sheet1.Range("H84" ).Left
      .Top = Sheet1.Range("H84" ).Top
      .IncrementLeft 20
      .IncrementTop 15
      .ZOrder msoSendToBack
    End With
    .Shapes("BackLabShape" ).ZOrder msoSendToBack
  End With
End Sub


Sub UngroupTemplateShape()
  On Error Resume Next
  With Sheet1.Shapes("TemplateGroup" )
    .Visible = msoCTrue
    .Ungroup
  End With
  On Error GoTo 0
End Sub
```

```
Option Explicit

Sub CreateLabelStep1()
  StopCalc 'Import Step 1
  GroupTemplateShapes
  ResetHideSteps
  Sheet1.Shapes("Step1" ).Fill.ForeColor.RGB = RGB(50, 255, 50)
  Sheet1.Shapes("Step1" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
  Sheet1.Range("5:16" ).EntireRow.Hidden = False
  Sheet1.Shapes("Step1Grp" ).Visible = msoCTrue
  On Error Resume Next
  Sheet1.Shapes("LabSheetPic" ).Visible = msoCTrue
  Sheet1.Range("E8" ).Select
  On Error GoTo 0
  ResetCalc
End Sub

Sub CreateLabelStep2()
  StopCalc 'Import Step 2
  GroupTemplateShapes
  ResetHideSteps
  Sheet1.Range("17:77" ).EntireRow.Hidden = False
  Sheet1.Shapes("Step2" ).Fill.ForeColor.RGB = RGB(50, 255, 50)
  Sheet1.Shapes("Step2" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
  Sheet1.Shapes("Step2Grp" ).Visible = msoCTrue
  Sheet1.Range("E20" ).Select
  ResetCalc
End Sub

Sub CreateLabelStep3()
  If Sheet1.Range("E20" ).Value = Empty Then
    MsgBox "Please select a data file for your labels before moving onto the next step"
    Exit Sub
  End If
  StopCalc 'Import Step 3
  ResetHideSteps
  Sheet1.Range("78:132" ).EntireRow.Hidden = False
  Sheet1.Shapes("Step3" ).Fill.ForeColor.RGB = RGB(50, 255, 50)
  Sheet1.Shapes("Step3" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
  Sheet1.Shapes("Step3Grp" ).Visible = msoCTrue
  UngroupTemplateShape 'Ungroup any lable shapes (if existing)
  With Sheet1.Shapes("BackLabShape" )
    .Height = Application.InchesToPoints(Sheet1.Range("E10" ).Value)
    .Width = Application.InchesToPoints(Sheet1.Range("G10" ).Value)
    .Visible = msoCTrue
  End With
  ImportLabelData
  ResetCalc
End Sub

Sub CreateLabelStep4()
  StopCalc 'Import Step 4
  GroupTemplateShapes 'Group Shapes before moving to Step 4
  ResetHideSteps
  Sheet1.Range("133:180" ).EntireRow.Hidden = False
  Sheet1.Shapes("Step4" ).Fill.ForeColor.RGB = RGB(50, 255, 50)
  Sheet1.Shapes("Step4" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
  Sheet1.Shapes("Step4Grp" ).Visible = msoCTrue
  BuildLabelSheet
  ResetCalc
  FilterDataCreateLabels
1
```

```
1
End Sub

Sub ResetHideSteps()
  With Sheet1 'Hide All Shapes & Rows
    .Range("5:180" ).EntireRow.Hidden = True
    .Shapes.Range(Array("ButtonSet" )).Fill.ForeColor.RGB = RGB(0, 97, 0)
    .Shapes.Range(Array("ButtonSet" )).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(
    255, 255, 255)
    .Shapes("Step1Grp" ).Visible = msoFalse
    .Shapes("Step2Grp" ).Visible = msoFalse
    .Shapes("Step3Grp" ).Visible = msoFalse
    .Shapes("Step4Grp" ).Visible = msoFalse
    On Error Resume Next
    Sheet1.Shapes("LabSheetPic" ).Visible = msoFalse 'Hide Label Sheet Icon
    Sheet1.Shapes("BackLabShape" ).Visible = msoFalse 'Hide LabelTemplate Background
    Sheet1.Shapes("LabPreview" ).Visible = msoFalse
    On Error GoTo 0
  End With
End Sub
```

Thank You!

This source code was created and made available to help you gain a better understanding of how VBA is used to create amazing Excel-based applications.

Thank you so much for your continued shares, likes and support. It really helps.

Excel For Freelancers