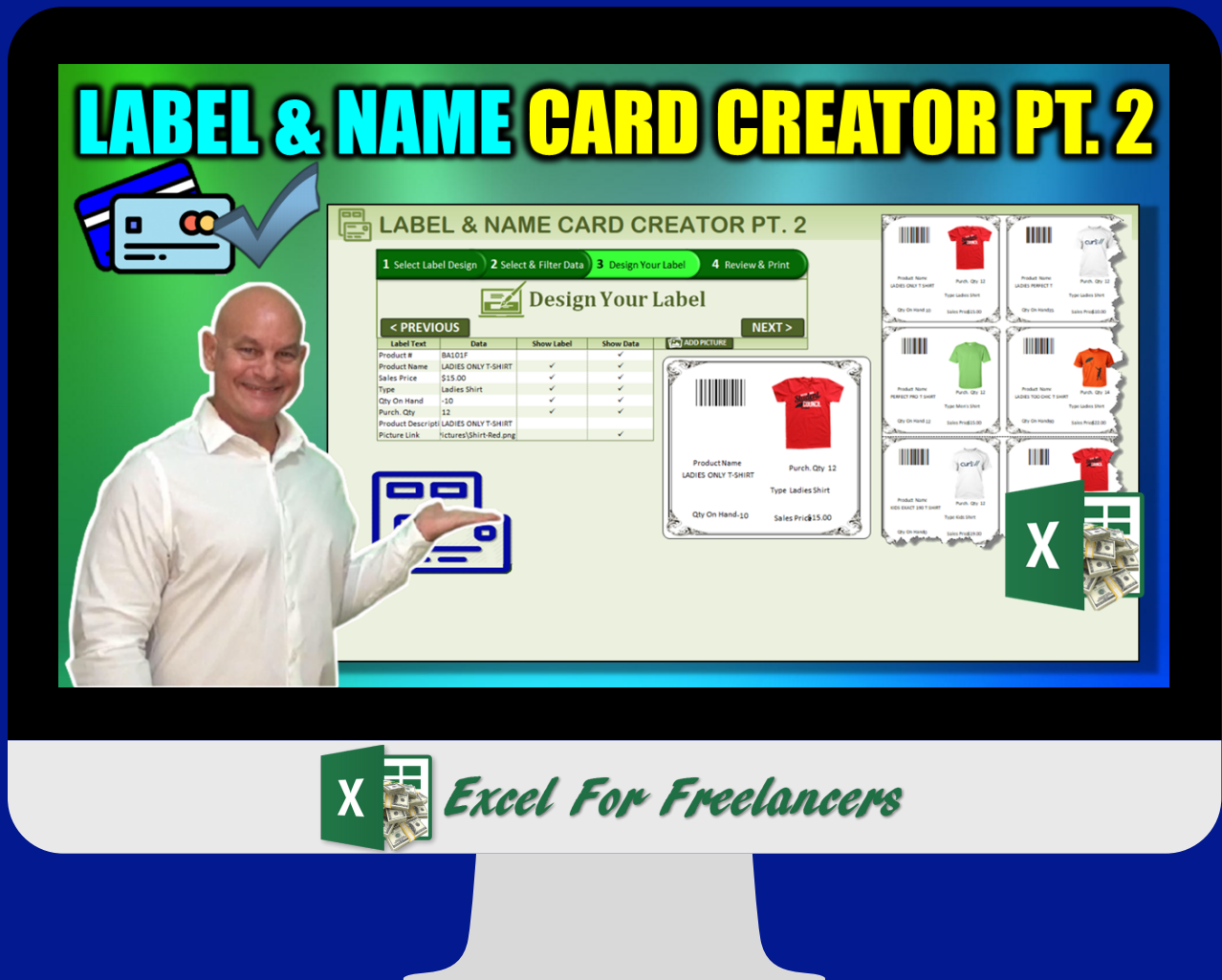


# VBA SOURCE CODE BOOK



**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](#)



**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](#)

Projects.....	2
VBAProject .....	2
Documents .....	2
Sheet1.....	2
(Declarations).....	2
Worksheet_Change [Sub ].....	2
Worksheet_SelectionChange [Sub ] .....	2
Sheet2.....	5
(Declarations).....	5
Sheet3.....	7
(Declarations).....	7
Sheet4.....	9
(Declarations).....	9
ThisWorkbook .....	11
(Declarations).....	11
Modules.....	13
CodeReset_Macs .....	13
(Declarations).....	13
CalculateRunTime_Minutes [Sub ] .....	13
ResetCalc [Sub ] .....	13
StartClock [Sub ] .....	13
StopCalc [Sub ] .....	13
StopClock [Sub ] .....	13
CreateLabelMacs .....	15
(Declarations).....	15
BuildLabelSheet [Sub ].....	15
FilterDataCreateLabels [Sub ] .....	16
PrintLabels [Sub ].....	19
DataWkBkMacros .....	21
(Declarations).....	21
BrowseForData [Sub ] .....	21
GetDataDetails [Sub ] .....	21
FilterBtnMacros .....	25
(Declarations).....	25
AddFilt_Numb_GreaterThan [Sub ] .....	25
AddFilt_Numb_LessThan [Sub ].....	25
AddFilt_Text_Contains [Sub ].....	25
AddFilt_Text_EndsWith [Sub ] .....	25
AddFilt_Text_StartsWith [Sub ] .....	25
ShowFilterGroup [Sub ] .....	26
Label_Load .....	28
(Declarations).....	28
LoadLabel [Sub ] .....	28
LabelDataMacros .....	30
(Declarations).....	30
ImportLabelData [Sub ] .....	30
LabelDesignMacs .....	33
(Declarations).....	33
Add_LabelPic [Sub ].....	33
GroupTemplateShapes [Sub ].....	33
Show_LabelPic [Sub ] .....	33
UngroupTemplateShape [Sub ] .....	34
TabMacros .....	36
(Declarations).....	36
CreateLabelStep1 [Sub ] .....	36
CreateLabelStep2 [Sub ] .....	36
CreateLabelStep3 [Sub ] .....	36
CreateLabelStep4 [Sub ] .....	36
ResetHideSteps [Sub ].....	37

## Option Explicit

```

1 Private Sub Worksheet_Change(ByVal Target As Range)
2
3   If Not Intersect(Target, Range("E8" )) Is Nothing Then
4     Range("B4" ).Value = True 'Set Template Change to True
5     LoadLabel 'Load Label details on change of label
6   End If
7   If Not Intersect(Target, Range("I21" )) Is Nothing Then
8     If Cells(Target.Row, Target.Column).Value <> Empty Then GetDataDetails 'Get
9       Destinationworksheet data on worksheet change
10    End If
11
12    'On Change of Column Format, when Column D is not empty, and not on Mapping load, then
13    Update Format Validation and Value
14    If Not Intersect(Target, Range("H27:H77" )) Is Nothing And Range("D" & Target.Row).Value
15    <> Empty Then
16      Range("I" & Target.Row).Validation.Delete
17      If Range("H" & Target.Row).Value <> "General" Then Range("I" & Target.Row).
18      Validation.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween,
19      Formula:="=" & Range("H" & Target.Row).Value & "FormatList"
20    End If
21
22    'On Change Of Format, update reformat sample Data
23    If Not Intersect(Target, Range("I27:I77" )) Is Nothing And Range("I" & Target.Row).Value
24    <> Empty Then
25      Range("E" & Target.Row).NumberFormat = Range("I" & Target.Row).Value 'Update Format
26      for sample data
27    End If
28  End Sub

```

```

29 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
30   If Target.CountLarge > 1 Then Exit Sub
31   Dim DesignRow As Long, DesignColumn As Long
32   Dim ShapeName As String, ShapeFormula As String, DataValue
33   If Target.CountLarge > 5 Then Exit Sub
34   'Display Filter Options on Selection
35   If Not Intersect(Target, Range("G27:G77" )) Is Nothing And Range("D" & Target.Row).Value
36   <> Empty And Target.Count < 2 Then
37     Range("B1" ).Value = Target.Row
38     ShowFilterGroup 'Display Text Filter Buttons
39   Else: 'Other than Filter Selection, hide filterGroup
40     Shapes("FiltGrp" ).Visible = msoFalse
41   End If
42
43   'Check/Uncheck Data & Label Options
44   If Not Intersect(Target, Range("F83:G132" )) Is Nothing And Range("D" & Target.Row).Value
45   <> Empty Then
46     DesignColumn = Target.Column 'Selected Design Column
47     DesignRow = Target.Row 'Selected Design Row
48     If DesignColumn = 6 Then ShapeName = "Label" & DesignRow Else ShapeName = "Data" &
49     DesignRow
50     If DesignColumn = 6 Then ShapeFormula = "=$D$" & DesignRow Else ShapeFormula = "=$E$"
51     & DesignRow
52     If DesignColumn = 7 Then DataValue = Range("E" & DesignRow).Value 'Get value of data
53     to check for possible pictures
54     If Target.Value = "ü" Then
55       Target.ClearContents
56       On Error Resume Next
57       Shapes(ShapeName).Delete 'Delete shape
58       If DesignColumn = 7 Then Shapes("ItemPicture" & DesignRow).Delete 'Deleete any
59       possible picture

```

```

49      1 2 3      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 = "ItemImage" & 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

```

**A**

Add, 2  
AlertStyle, 2

**C**

Cells, 2  
ClearContents, 2  
Column, 2  
Copy, 3  
Count, 2  
CountLarge, 2

**D**

DataValue, 2, 3  
Delete, 2  
DesignColumn, 2  
DesignRow, 2, 3  
Dir, 3  
DrawingObject, 3

**E**

Empty, 2  
Explicit, 2

**F**

Formula, 3  
Formula1, 2

**G**

GetDataDetails, 2

**H**

Height, 3

**I**

IncrementLeft, 3  
IncrementTop, 3  
Insert, 3  
InStr, 3  
Intersect, 2

**L**

Left, 3  
LoadLabel, 2  
LockAspectRatio, 3

**M**

MsgBox, 3  
msoCTrue, 3  
msoFalse, 2  
msoTrue, 3

**N**

Name, 3  
NumberFormat, 2

**O**

Operator, 2

**P**

Paste, 3  
Pictures, 3  
Placement, 3

**R**

Range, 2, 3  
Row, 2, 3

**S**

Selection, 3  
ShapeFormula, 2, 3  
ShapeName, 2, 3  
ShapeRange, 3  
Shapes, 2, 3  
Sheet1, 3  
ShowFilterGroup, 2

**T**

Target, 2, 3  
Top, 3

**V**

Validation, 2  
Value, 2, 3  
vbDirectory, 3  
Visible, 2, 3

**W**

Worksheet\_Change, 2  
Worksheet\_SelectionChange, 2

**X**

xlBetween, 2  
xlFreeFloating, 3  
xlValidAlertStop, 2  
xlValidateList, 2

1 Option Explicit

2



**E**

Explicit, 5

1 Option Explicit  
2

**E**

Explicit, 7

1	Option Explicit
2	

**E**

Explicit, 9

1	Option Explicit
2	

**E**

Explicit, [11](#)

```
1 Option Explicit
2 Dim StartTime As Double
3 Dim SecondsElapsed As Double
4
5 Sub CalculateRunTime_Minutes()
6 'PURPOSE: Determine how many minutes it took for code to completely run
7 'SOURCE: www.TheSpreadsheetGuru.com/the-code-vault
8
9 Dim StartTime As Double
10 Dim MinutesElapsed As String
11
12 'Remember time when macro starts
13 StartTime = Timer
14
15 '*****
16 'Insert Your Code Here...
17 '*****
18
19 'Determine how many seconds code took to run
20 MinutesElapsed = Format((Timer - StartTime) / 86400, "hh:mm:ss" )
21
22 'Notify user in seconds
23 MsgBox "This code ran successfully in " & MinutesElapsed & " minutes" , vbInformation
24
25 End Sub
26
27 Sub ResetCalc()
28 With Application
29 .EnableEvents = True
30 .Calculation = xlCalculationAutomatic
31 .ScreenUpdating = True
32 End With
33 End Sub
34
35
36
37 Sub StartClock()
38
39 'Remember time when macro starts
40 StartTime = Timer
41 End Sub
42
43 Sub StopCalc()
44 With Application
45 .EnableEvents = False
46 .Calculation = xlCalculationManual
47 .ScreenUpdating = False
48 End With
49 End Sub
50
51 Sub StopClock()
52
53 'Determine how many seconds code took to run
54 SecondsElapsed = Round(Timer - StartTime, 2)
55
56 'Notify user in seconds
57 MsgBox "This code ran successfully in " & SecondsElapsed & " seconds" , vbInformation
58 End Sub
```



**A**

Application, 13

**C**

CalculateRunTime\_Minutes, 13

Calculation, 13

**E**

EnableEvents, 13

Explicit, 13

**F**

Format, 13

**M**

MinutesElapsed, 13

MsgBox, 13

**R**

ResetCalc, 13

Round, 13

**S**

ScreenUpdating, 13

SecondsElapsed, 13

StartClock, 13

StartTime, 13

StopCalc, 13

StopClock, 13

**T**

Timer, 13

**V**

vbInformation, 13

**X**

xlCalculationAutomatic, 13

xlCalculationManual, 13

```

1 Option Explicit
2 Sub BuildLabelSheet()
3 'If Sheet1.Range("B4").Value = False Then Exit Sub 'No need to rebuild label sheet unless
  template has changed
4 StopCalc
5 Dim LabSheetQty As Long, LabelCol As Long, LabelRow As Long, SetCount As Long
6 Dim LastLabelRow As Long, LastLabelCol As Long, SetCol As Long, SetRow As Long
7 Dim DataRow As Long, LastDataRow, MaxLabels As Long, MaxRows As Long
8 Dim LabHeight As Double, LabWidth As Double, ColumnSpace As Double, RowSpace As Double
9 Dim LeftMargin As Double, RightMargin As Double, TopMargin As Double, BottomMargin As
  Double
10 Dim PrintArea As Range
11 With Sheet1
12   LabSheetQty = .Range("E9").Value 'Label Sheet Qty
13   LabelCol = .Range("E11").Value 'Number of label columns
14   LabelRow = .Range("G11").Value 'Number of label rows
15   LabHeight = .Range("E10").Value 'Label Height
16   LabWidth = .Range("G10").Value 'Label Width
17   ColumnSpace = .Range("E13").Value 'Column Spacing
18   RowSpace = .Range("G13").Value 'Row Spacing
19   LeftMargin = .Range("E14").Value 'Left Margin Spacing
20   RightMargin = .Range("G14").Value 'Right Margin
21   TopMargin = .Range("E15").Value 'Top Margin
22   BottomMargin = .Range("G15").Value 'Bottom Margin
23   MaxLabels = .Range("H22").Value - 1 'Maximum # of Labels (without any filter)
24   MaxRows = MaxLabels / LabelCol
25   'Determine the Total # of Rows (Full Sheets) Round Up To Whole # (Maximum Rows / Rows
  Per sheet) * Number OF Rows Per Sheet
26   LastLabelRow = Application.WorksheetFunction.RoundUp((MaxRows / LabelRow), 0) *
  LabelRow
27   LastLabelCol = LabelCol + (LabelCol - 1)
28   LastLabelRow = LastLabelRow + (LastLabelRow - 1) 'Set the Last Label Row (Use full
  Sheets)
29 End With
30 With Sheet3 'Label Template
31   'Setup Columns
32   For SetCol = 1 To LastLabelCol Step 2
33     For SetCount = 1 To 3 'Run 3 times to set width
34       .Cells(1, SetCol).ColumnWidth = Application.InchesToPoints(LabWidth) * (.Cells(1,
  SetCol).ColumnWidth / .Cells(1, SetCol).Width) - 3
35     Next SetCount
36     If SetCol <> LastLabelCol Then .Cells(1, SetCol + 1).ColumnWidth = ColumnSpace
37   Next SetCol
38   'Setup Rows
39   For SetRow = 1 To LastLabelRow Step 2
40     .Cells(SetRow, 1).RowHeight = Application.InchesToPoints(LabHeight) 'Set Row Height
41     .Cells(SetRow + 1, 1).RowHeight = RowSpace
42   Next SetRow
43   With .PageSetup
44     .PrintArea = Range(Sheet3.Cells(1, 1), Sheet3.Cells(LastLabelRow, LastLabelCol)).
  Address
45     .LeftMargin = Application.InchesToPoints(LeftMargin)
46     .RightMargin = Application.InchesToPoints(RightMargin)
47     .TopMargin = Application.InchesToPoints(TopMargin)
48     .BottomMargin = Application.InchesToPoints(BottomMargin)
49     .HeaderMargin = 0
50     .FooterMargin = 0
51   End With
52   Sheet1.Range("B4").Value = False 'Reset Label Change to False
53 End With

```

```

1
54     ResetCalc
55 End Sub
56 Sub FilterDataCreateLabels()
57     Dim FirstFiltCol As Long, LastFiltCol As Long, TotalDataCol As Long
58     Dim LabelCol As Long, LabelRow As Long, LabelNumber As Long
59     Dim TotalLabelCol As Long, TotalLabelRow As Long
60     Dim FirstLabelRow As Long, LastLabelRow As Long, LabelDataRow As Long
61     Dim LastLabelCol As Long, LabelDataCol As Long
62     Dim DesignRow As Long, LastDesignRow As Long
63     Dim FirstDataRow As Long, LastDataRow As Long, LastHeaderRow As Long
64     Dim LabelTemplate As Shape, TempLabelShape As Shape, LabelShape As Shape, ItemPic As Shape

65     Dim LabelRng As Range, LabelCell As Range
66     Dim PicFileName As String
67     'Remove any labels currently on the sheet
68     For Each LabelShape In Sheet3.Shapes
69         LabelShape.Delete
70     Next
71     Set LabelShape = Nothing
72     'Remove any Label Preview (if existing)
73     On Error Resume Next
74     Sheet1.Shapes("LabPreview").Delete
75     On Error GoTo 0

76
77     With Sheet1
78         TotalDataCol = .Range("J23").Value
79         FirstFiltCol = TotalDataCol + 5
80         LastFiltCol = FirstFiltCol + TotalDataCol
81         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
82         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
83         LastHeaderRow = Application.WorksheetFunction.CountA(.Range("D27:D77")) + 26 'Last
            header Row
84         If LastHeaderRow < 27 Then Exit Sub 'On No Headers
85         .Range("G27:G" & LastHeaderRow).Copy
86         Sheet4.Cells(2, FirstFiltCol).PasteSpecial xlPasteValues, , , True 'Copy Over filters
87         LastDataRow = .Range("H22").Value - 1 'Last Row Of Data
88         Range(Sheet4.Cells(1, 1), Sheet4.Cells(LastDataRow, TotalDataCol + 1)).AdvancedFilter
            xlFilterCopy, _
89         CriteriaRange:=Range(Sheet4.Cells(1, FirstFiltCol), Sheet4.Cells(2, LastFiltCol)),
            CopyToRange:=Range(Sheet4.Cells(3, FirstFiltCol), Sheet4.Cells(3, LastFiltCol)), Unique
            :=True

90
91         'Builld Labels On Label Page
92         On Error Resume Next
93         Set LabelTemplate = Sheet1.Shapes("TemplateGroup")
94         On Error GoTo 0
95         If LabelTemplate Is Nothing Then
96             GroupTemplateShapes 'Run Macro to Create Label Group
97             On Error Resume Next
98             Set LabelTemplate = Sheet1.Shapes("TemplateGroup") '2nd ry
99             On Error GoTo 0
100        End If
101        If LabelTemplate Is Nothing Then Exit Sub 'Exit on 2nd Failure to set locate label
            template
102        LastDataRow = Sheet4.Cells(99999, FirstFiltCol).End(xlUp).Row 'Last Results Row
103        .Range("E141").Value = LastDataRow - 3 'Display # Of labels to Print
104        .Range("G141").Value = Application.WorksheetFunction.RoundUp((LastDataRow - 3) / .
            Range("E9").Value, 0) 'Total # Of Labels Divided by labels per sheet

```

1 2

```

1 2
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)

```

```

1 2 3 4 5 6 7 8 9
157 .Height = ItemPic.Height 'Set Height same as original picture
158 .Width = ItemPic.Width 'Set width same as original picture
159 .Left = Sheet3.Cells(LabelRow, LabelCol).Left 'Set initial Left
    position based on cell
160 .Top = Sheet3.Cells(LabelRow, LabelCol).Top 'Set initial Top
    Position based on cell
161 .IncrementLeft ItemPic.Left - Sheet1.Shapes("BackLabShape" ).Left
    'Increment left (move to right) based on distance from Label
    background and picture
162 .IncrementTop ItemPic.Top - Sheet1.Shapes("BackLabShape" ).Top
    'Increment top (move down) based on distance from Label background
    and picture
163 .Visible = msoCTrue 'Display Picture
164 End With
165
166 End If
167
168 Else ' Not Picture
169     'On Error Resume Next
170
171     Sheet3.Shapes("Data" & DesignRow).DrawingObject.Formula = "=" & Imported
    Data!" & Sheet4.Cells(LabelDataRow, LabelDataCol).Address 'Get Data
    Value
172     Sheet3.Shapes("Data" & DesignRow).Name = "Label" & LabelNumber &
    "_Data" & DesignRow
173     'On Error GoTo 0
174     Sheet1.Shapes("Data" & DesignRow).PickUp
175     Sheet3.Shapes("Label" & LabelNumber & "_Data" & DesignRow).Apply
176 End If
177 End If
178 On Error Resume Next
179 Sheet3.Shapes("ItemPicture" & DesignRow).Delete 'Remove Template picture
    copied over
180 On Error GoTo 0
181 LabelDataCol = LabelDataCol + 1 'Increment Data Column
182
183 Next DesignRow
184
185
186 Set TempLabelShape = Sheet3.Shapes("TemplateGroup" ) 'Reset Temp Label To
    Original sheet Template
187
188 LabelNumber = LabelNumber + 1
189 LabelDataRow = LabelDataRow + 1
190 If LabelDataRow > LastDataRow Then Exit For
191 Next LabelCol
192 If LabelDataRow > LastDataRow Then Exit For
193 Next LabelRow
194
195
196 On Error Resume Next
197 Sheet3.Shapes("TemplateGroup" ).Delete 'Remove temporary group
198 On Error GoTo 0
199 If LabelRow > 10 Then LabelRow = 10 'Set max height for sample picture
200 Set LabelRng = Range(Sheet3.Cells(1, 1), Sheet3.Cells(LabelRow, LastLabelCol))
201 Application.CutCopyMode = False
202 Application.Wait Now + 0.00001
203 LabelRng.CopyPicture
204 Sheet1.Activate
205 Application.Wait Now + 0.00001
206 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
```

—, 16

## A

Activate, 17, 18  
 Address, 15, 18, 19  
 AdvancedFilter, 16  
 Application, 15-19  
 Apply, 18

## B

BottomMargin, 15  
 BuildLabelSheet, 15

## C

Cells, 15-18  
 ColumnSpace, 15  
 ColumnWidth, 15  
 Copy, 16, 17, 19  
 CopyPicture, 18  
 CopyToRange, 16  
 CountA, 16  
 CriteriaRange, 16  
 CutCopyMode, 18, 19

## D

DataRow, 15  
 Delete, 16, 18  
 DesignRow, 16-18  
 Destination, 17, 19  
 DrawingObject, 18

## E

ErrorHandler, 17, 19  
 Explicit, 15

## F

FilterDataCreateLabels, 16  
 FirstDataRow, 16  
 FirstFiltCol, 16, 17  
 FirstLabelRow, 16  
 FooterMargin, 15  
 Formula, 18, 19

## G

GroupTemplateShapes, 16

## H

HeaderMargin, 15  
 Height, 18

## I

InchesToPoints, 15  
 IncrementLeft, 18, 19  
 IncrementTop, 18  
 Insert, 17  
 InStr, 17  
 ItemPic, 16-18

## L

LabelCell, 16, 17, 19  
 LabelCol, 15-18  
 LabelDataCol, 16-18  
 LabelDataRow, 16-18  
 LabelNumber, 16-18  
 LabelRng, 16, 18, 19  
 LabelRow, 15-18  
 LabelShape, 16  
 LabelTemplate, 16, 17

LabHeight, 15  
 LabSheetQty, 15  
 LabWidth, 15  
 LastDataRow, 15-18  
 LastDesignRow, 16, 17  
 LastFiltCol, 16  
 LastHeaderRow, 16, 17  
 LastLabelCol, 15-18  
 LastLabelRow, 15-17  
 Left, 17, 18  
 LeftMargin, 15

## M

MaxLabels, 15  
 MaxRows, 15  
 msoCTrue, 17, 18

## N

Name, 17-19  
 Now, 18, 19

## P

PageSetup, 15  
 Paste, 17, 19  
 PasteSpecial, 16, 18  
 PicFileName, 16  
 PickUp, 18  
 Pictures, 17  
 PrintArea, 15  
 PrintLabels, 19  
 PrintOut, 19

## R

Range, 15-19  
 ResetCalc, 16  
 RightMargin, 15  
 RoundUp, 15-17  
 Row, 16  
 RowHeight, 15  
 RowSpace, 15

## S

Selection, 19  
 SetCol, 15  
 SetCount, 15  
 SetRow, 15  
 Shape, 16  
 ShapeRange, 17  
 Shapes, 16-19  
 Sheet1, 15-19  
 Sheet3, 15-19  
 Sheet4, 16-18  
 Sheets, 17  
 StopCalc, 15

## T

TempLabelShape, 16-19  
 ThisWorkbook, 17  
 Top, 17, 18  
 TopMargin, 15  
 TotalDataCol, 16  
 TotalLabelCol, 16, 17  
 TotalLabelRow, 16, 17

## U

Unique, 16

## V

Value, 15-17  
 Visible, 17, 18

## W

Wait, 18, 19  
 Width, 15, 18  
 WorksheetFunction, 15-17

## X

xlFilterCopy, 16  
 xlPasteValues, 16  
 xlUp, 16

```

1  Option Explicit
2
3  Sub BrowseForData()
4      Dim FileFldr As FileDialog
5      Dim DataFilePath As String
6      Dim DataWs As Worksheet
7      Dim ThisWkBk As Workbook, DataWkBk As Workbook
8      Dim WsRow As Long
9
10     Set ThisWkBk = ThisWorkbook
11     With Sheet1
12         WsRow = 20
13         .Range("A20:A50,E20:H20,I21:J21,F22:F23,H22:H23").ClearContents 'Clear Any older data
14         .Range("D27:J77").ClearContents 'Clear any older Worksheet data
15         Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
16         With FileFldr
17             .Title = "Select Data File"
18             .Filters.Add "Excel Files", "*.xlsm,*.xlsx,*.xlsb,*.xls"
19             If .Show <> -1 Then GoTo NoSelection
20             Sheet1.Range("E20").Value = .SelectedItems(1) 'Get Full File Path
21         End With
22
23         DataFilePath = .Range("E20").Value 'Long File Path
24         .Range("E21").Value = Dir(DataFilePath, vbDirectory) 'File Name
25         Workbooks.Open (DataFilePath) 'Open Data Workbook
26         Set DataWkBk = Workbooks(Dir(DataFilePath)) 'Set Data Workbook
27
28         For Each DataWs In DataWkBk.Worksheets
29             Sheet1.Range("A" & WsRow).Value = DataWs.Name
30             WsRow = WsRow + 1
31         Next
32         ThisWkBk.Activate
33         .Range("I21").Select 'Select cell for user to select sheet
34         NoSelection:
35     End With
36 End Sub
37
38
39 Sub GetDataDetails()
40     Dim DataFilePath As String, DataShtName As String
41     Dim WsRow As Long, FirstRow As Long, TableRow As Long, TableCol As Long, FirstCol As Long,
42     LastRow As Long, LastCol As Long
43     Dim DataRow As Long, DataCol As Long, DesignRow As Long
44     Dim DataWkBk As Workbook, ThisWkBk As Workbook
45
46     Set ThisWkBk = ThisWorkbook
47     With Sheet1
48         DataFilePath = .Range("E20").Value 'Data Workbook File Path
49         On Error GoTo MissingWorkbook
50         Set DataWkBk = Workbooks(Dir(DataFilePath)) 'Set Data Workbook
51         On Error GoTo 0
52         DataShtName = .Range("I21").Value 'Data Worksheet Name
53         .Range("D27:J77").ClearContents 'Clear any older Worksheet data
54         .Range("D83:G132").ClearContents 'Clear any Lables & Data for design steps
55     End With
56
57     With DataWkBk.Sheets(DataShtName)
58         'Get Last Row Of Data
59

```



```

1 2
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,
71         LastCol))) / LastCol) >= 0.7 Then
72         FirstRow = TableRow
73         Exit For
74     End If
75 Next TableRow
76
77 'Determine First Column Of Table w/ 50% of the cells filled
78 For TableCol = 1 To LastCol
79     If (Application.WorksheetFunction.CountA(Range(.Cells(1, TableCol), .Cells(LastRow,
80         TableCol))) / LastRow) >= 0.5 Then
81         FirstCol = TableCol
82         Exit For
83     End If
84 Next TableCol
85
86 End With
87
88 If Sheet1.Range("B18").Value = True Then FirstRow = FirstRow + 1 'Has Headers
89
90 Sheet1.Range("F22").Value = FirstRow 'Set First Row Of Data Data
91 Sheet1.Range("H22").Value = LastRow 'Set Last Row Of Data Data
92 Sheet1.Range("F23").Value = FirstCol 'Set First Column Of Data Data
93 Sheet1.Range("H23").Value = LastCol 'Set Last Column Of Data Data
94
95 ThisWkBk.Activate
96 'Load Data Details
97
98 DataRow = 27
99 DesignRow = 83
100 With Sheet1
101     If .Range("B18").Value = True Then 'Data has Headers
102         For DataCol = FirstCol To LastCol
103             .Range("D" & DataRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow - 1,
104                 DataCol).Value 'Load Headers
105             .Range("D" & DesignRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow - 1,
106                 DataCol).Value 'Load Headers
107             DataRow = DataRow + 1
108             DesignRow = DesignRow + 1
109         Next DataCol
110     End If
111     'Add First Row Of Data
112     DataRow = 27
113     DesignRow = 83
114     For DataCol = FirstCol To LastCol
115         .Range("E" & DataRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow, DataCol)
116         .Value
117         .Range("E" & DesignRow).Value = DataWkBk.Sheets(DataShtName).Cells(FirstRow,

```

1 2 3

```
1 2 3
114 DataCol).Value 'Load First Row OF Data
115 DataRow = DataRow + 1
116 DesignRow = DesignRow + 1
117 Next DataCol
118 On Error Resume Next
119 Sheet1.Shapes("TemplateGroup" ).Delete 'Remove all data & label fields from any
120 previous label
121 On Error GoTo 0
122 End With
123 Exit Sub
124 MissingWorkbook:
125 MsgBox "Please check for a correct Data Workbook"
126 End Sub
127
```

—  
→, 22

## A

Activate, 21, 22  
Add, 21  
After, 22  
Application, 21, 22

## B

BrowseForData, 21

## C

Cells, 22  
ClearContents, 21  
Column, 22  
CountA, 22

## D

DataCol, 21-23  
DataFilePath, 21  
DataRow, 21-23  
DataShtName, 21, 22  
DataWkBk, 21, 22  
DataWs, 21  
Delete, 23  
DesignRow, 21-23  
Dir, 21

## E

Explicit, 21

## F

FileDialog, 21  
FileFldr, 21  
Filters, 21  
Find, 22  
FirstCol, 21, 22  
FirstRow, 21, 22

## G

GetDataDetails, 21

## L

LastCol, 21, 22  
LastRow, 21, 22  
Lookat, 22  
LookIn, 22

## M

MatchCase, 22  
MissingWorkbook, 21, 23  
MsgBox, 23  
msoFileDialogFilePicker, 21

## N

Name, 21  
NoSelection, 21

## O

Open, 21

## R

Range, 21, 22  
Row, 22

## S

SearchDirection, 22  
SearchOrder, 22

SelectedItems, 21

Shapes, 23  
Sheet1, 21-23  
Sheets, 21, 22  
Show, 21

## T

TableCol, 21, 22  
TableRow, 21, 22  
ThisWkBk, 21, 22  
ThisWorkbook, 21  
Title, 21

## V

Value, 21-23  
vbDirectory, 21

## W

What, 22  
Workbook, 21  
Workbooks, 21  
Worksheet, 21  
WorksheetFunction, 22  
Worksheets, 21  
WsRow, 21

## X

xlByColumns, 22  
xlByRows, 22  
xlFormulas, 22  
xlPart, 22  
xlPrevious, 22

```
1 Option Explicit
2
3
4 Sub AddFilt_Numb_GreaterThan()
5     Dim DataRow As Long
6     Dim FilterBy As String
7     With Sheet1
8         If .Range("B1").Value = Empty Then Exit Sub
9         DataRow = .Range("B1").Value
10        If .Range("G" & DataRow).Value = Empty Then Exit Sub
11        FilterBy = .Range("G" & DataRow).Value 'Filter By Text
12        FilterBy = Replace(Replace(Replace(FilterBy, "*", "" ), "<" , "" ), ">" , "" )
13        'Remove any previous filtering characters
14        .Range("G" & DataRow).Value = ">" & FilterBy
15    End With
16 End Sub
17
18 Sub AddFilt_Numb_LessThan()
19     Dim DataRow As Long
20     Dim FilterBy As String
21     With Sheet1
22         If .Range("B1").Value = Empty Then Exit Sub
23         DataRow = .Range("B1").Value
24         If .Range("G" & DataRow).Value = Empty Then Exit Sub
25         FilterBy = .Range("G" & DataRow).Value 'Filter By Text
26         FilterBy = Replace(Replace(Replace(FilterBy, "*", "" ), "<" , "" ), ">" , "" )
27         'Remove any previous filtering characters
28         .Range("G" & DataRow).Value = "<" & FilterBy
29    End With
30 End Sub
31
32 Sub AddFilt_Text_Contains()
33     Dim DataRow As Long
34     Dim FilterBy As String
35     With Sheet1
36         If .Range("B1").Value = Empty Then Exit Sub
37         DataRow = .Range("B1").Value
38         If .Range("G" & DataRow).Value = Empty Then Exit Sub
39         FilterBy = .Range("G" & DataRow).Value 'Filter By Text
40         FilterBy = Replace(Replace(Replace(FilterBy, "*", "" ), "<" , "" ), ">" , "" )
41         'Remove any previous filtering characters
42         .Range("G" & DataRow).Value = "*" & FilterBy & "*"
43    End With
44 End Sub
45
46 Sub AddFilt_Text_EndsWith()
47     Dim DataRow As Long
48     Dim FilterBy As String
49     With Sheet1
50         If .Range("B1").Value = Empty Then Exit Sub
51         DataRow = .Range("B1").Value
52         If .Range("G" & DataRow).Value = Empty Then Exit Sub
53         FilterBy = .Range("G" & DataRow).Value 'Filter By Text
54         FilterBy = Replace(Replace(Replace(FilterBy, "*", "" ), "<" , "" ), ">" , "" )
55         'Remove any previous filtering characters
56         .Range("G" & DataRow).Value = "*" & FilterBy
57    End With
58 End Sub
59
60 Sub AddFilt_Text_StartsWith()
61     Dim DataRow As Long
62     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
```

## A

AddFilt\_Numb\_GreaterThan, [25](#)  
AddFilt\_Numb\_LessThan, [25](#)  
AddFilt\_Text\_Contains, [25](#)  
AddFilt\_Text\_EndsWith, [25](#)  
AddFilt\_Text\_StartsWith, [25](#)

## D

DataRow, [25](#), [26](#)

## E

Empty, [25](#), [26](#)  
Explicit, [25](#)

## F

FilterBy, [25](#), [26](#)

## I

IncrementTop, [26](#)

## L

Left, [26](#)

## M

msoCTrue, [26](#)

## R

Range, [25](#), [26](#)  
Replace, [25](#), [26](#)

## S

Shapes, [26](#)  
Sheet1, [25](#), [26](#)  
ShowFilterGroup, [26](#)

## T

Top, [26](#)

## V

Value, [25](#), [26](#)  
Visible, [26](#)

```
1 Option Explicit
2
3
4 Sub LoadLabel()
5     With Sheet1
6         Dim LabelRow As Long
7         Dim PicName As String
8         If .Range("B5").Value = Empty Then
9             MsgBox "Please select a correct label from the drop down list"
10            Exit Sub
11        End If
12        LabelRow = .Range("B5").Value 'Label Row
13        .Range("E9").Value = Sheet2.Range("B" & LabelRow).Value '# Labels Per Sheet
14        .Range("G9").Value = Sheet2.Range("C" & LabelRow).Value 'Type
15        .Range("E10").Value = Sheet2.Range("D" & LabelRow).Value 'Height
16        .Range("G10").Value = Sheet2.Range("E" & LabelRow).Value 'Width
17        .Range("E11").Value = Sheet2.Range("F" & LabelRow).Value 'Columns
18        .Range("G11").Value = Sheet2.Range("G" & LabelRow).Value 'Rows
19        .Range("E13").Value = Sheet2.Range("I" & LabelRow).Value 'Column Spacing
20        .Range("G13").Value = Sheet2.Range("J" & LabelRow).Value 'Row Spacing
21        .Range("E14").Value = Sheet2.Range("K" & LabelRow).Value 'Left Margin
22        .Range("G14").Value = Sheet2.Range("L" & LabelRow).Value 'Right Margin
23        .Range("E15").Value = Sheet2.Range("M" & LabelRow).Value 'Top Margin
24        .Range("G15").Value = Sheet2.Range("N" & LabelRow).Value 'Bottom Margin
25        PicName = Sheet2.Range("H" & LabelRow).Value 'Filename
26        Sheet2.Shapes(PicName).Copy
27        .Range("H8").Select
28        .Paste
29        On Error Resume Next
30        .Shapes("LabSheetPic").Delete
31        On Error GoTo 0
32        With .Shapes(PicName)
33            .Name = "LabSheetPic"
34            .Left = Sheet1.Range("H8").Left
35            .Top = Sheet1.Range("H8").Top
36            .LockAspectRatio = msoCTrue
37            .Width = 90
38            .IncrementLeft 40
39        End With
40        .Range("E8").Select
41    End With
End Sub
```

## C

Copy, [28](#)

## D

Delete, [28](#)

## E

Empty, [28](#)

Explicit, [28](#)

## I

IncrementLeft, [28](#)

## L

LabelRow, [28](#)

Left, [28](#)

LoadLabel, [28](#)

LockAspectRatio, [28](#)

## M

MsgBox, [28](#)

msoCTrue, [28](#)

## N

Name, [28](#)

## P

Paste, [28](#)

PicName, [28](#)

## R

Range, [28](#)

## S

Shapes, [28](#)

Sheet1, [28](#)

Sheet2, [28](#)

## T

Top, [28](#)

## V

Value, [28](#)

## W

Width, [28](#)



```

1  Option Explicit
2
3  Sub ImportLabelData()
4      Dim DataFilePath As String, DataShtName As String
5      Dim DataWkBk As Workbook, ThisWkBk As Workbook
6
7      Dim FirstDataRow As Long, FirstDataCol As Long, LastDataCol As Long
8      Dim LastDataRow As Long, DataCol As Long, DataRow As Long
9
10     Dim TotalDataRows As Long, TotalDataColumns As Long
11     Dim TableRow As Long
12     Dim TableCol As Long
13     Dim ColCount As Long
14     Dim FirstAvailRow As Long
15
16
17     Dim MapCol As Long
18     Dim LastMapCol As Long
19     Dim CellFormat As String
20
21     With Sheet1
22         'Make sure an proper file path
23         DataFilePath = .Range("E20").Value 'Date File Path
24         If Dir(DataFilePath) = "" Then
25             MsgBox "Please select a proper data file for your labels"
26             CreateLabelStep2
27             Exit Sub
28         End If
29
30         Set ThisWkBk = ThisWorkbook 'Set Current Workbook To Import
31
32         'Get Data Workbook Variables
33         On Error Resume Next 'An error will occur if we set a workbook that is not open. If
34         the error occurs we can run a check and open the workbook
35         Set DataWkBk = Workbooks(Dir(DataFilePath)) 'Set Data Workbook
36         If DataWkBk Is Nothing Then 'Data Workbook is not open
37             Set DataWkBk = Workbooks.Open(DataFilePath)
38         End If
39         DataShtName = .Range("I21").Value 'Data Worksheet Name
40         FirstDataRow = .Range("F22").Value 'First Row of Data (excludes headers)
41         FirstDataCol = .Range("F23").Value 'First Column Of Data from the Data sheet
42         LastDataCol = .Range("H23").Value 'Last Data Column
43         LastDataRow = .Range("H22").Value 'Last Row Of Data
44         TotalDataRows = .Range("J22").Value 'Total Rows To Be copied
45         TotalDataColumns = .Range("J23").Value 'Total Columns To be copied
46         Sheet4.Range("A1:ZZ9999").ClearContents 'Clear Any Existing Data
47
48         'Copy Over headers (if any)
49         If .Range("B18").Value = True Then Range(Sheet4.Cells(1, 1), Sheet4.Cells(1,
50         TotalDataColumns)).Value = _
51         Range(DataWkBk.Sheets(DataShtName).Cells(FirstDataRow - 1, FirstDataCol), DataWkBk.
52         Sheets(DataShtName).Cells(FirstDataRow - 1, LastDataCol)).Value
53
54         'Copy Over Data
55         Range(Sheet4.Cells(2, 1), Sheet4.Cells(TotalDataRows + 1, TotalDataColumns)).Value = _
56         Range(DataWkBk.Sheets(DataShtName).Cells(FirstDataRow, FirstDataCol), DataWkBk.Sheets(
57         DataShtName).Cells(LastDataRow, LastDataCol)).Value
58         ThisWorkbook.Activate
59         Exit Sub
60     DuplicateSheet:
61     MsgBox "The Sheet Name '" & .Range("I30").Value & "' already exists" & vbCrLf &

```

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

—  
→, 30

## A

Activate, 30

## C

CellFormat, 30

Cells, 30

ClearContents, 30

ColCount, 30

CreateLabelStep2, 30

## D

DataCol, 30

DataFilePath, 30

DataRow, 30

DataShtName, 30

DataWkBk, 30

Dir, 30

DuplicateSheet, 30

## E

Explicit, 30

## F

FirstAvailRow, 30

FirstDataCol, 30

FirstDataRow, 30

## I

ImportLabelData, 30

## L

LastDataCol, 30

LastDataRow, 30

LastMapCol, 30

## M

MapCol, 30

MsgBox, 30

## O

Open, 30

## R

Range, 30

## S

Sheet1, 30

Sheet4, 30

Sheets, 30

## T

TableCol, 30

TableRow, 30

ThisWkBk, 30

ThisWorkbook, 30

TotalDataColumns, 30

TotalDataRows, 30

## V

Value, 30

vbCrLf, 30

## W

Workbook, 30

Workbooks, 30

```

1  Option Explicit
2
3  Sub Add_LabelPic()
4      Dim PicFile As FileDialog
5      With Sheet1
6          Set PicFile = Application.FileDialog(msoFileDialogFilePicker)
7          With PicFile
8              .Title = "Select an Employee Picture"
9              .Filters.Add "All Picture Files" , "*.jpg,*jpeg,*gif*.png*.gif*.bmp*.tiff" , 1
10             If .Show <> -1 Then GoTo NoSelection
11             Sheet1.Range("B84" ).Value = .SelectedItems(1) 'Put File Name in B84
12         End With
13         'Run Macro to Show Picture
14         Show_LabelPic
15     NoSelection:
16 End With
17 End Sub
18
19 Sub GroupTemplateShapes()
20     With Sheet1
21         Dim LabelShape As Shape, TemplateGroup As Shape
22         Dim ShapesNames As String
23         Dim GrpArr As Variant
24         For Each LabelShape In Sheet1.Shapes
25             If InStr(LabelShape.Name, "Data" ) > 0 Or InStr(LabelShape.Name, "Label" ) > 0 Or
                InStr(LabelShape.Name, "LabelPic" ) Or InStr(LabelShape.Name, "ItemPicture" ) > 0
26             Then
27                 ShapesNames = ShapesNames & LabelShape.Name & ","
28             End If
29         Next LabelShape
30         If ShapesNames = Empty Then Exit Sub 'Grouped already or no shapes exist
31         ShapesNames = Left(ShapesNames, Len(ShapesNames) - 1)
32         GrpArr = Split(ShapesNames, "," )
33         On Error GoTo NoGroup
34         Set TemplateGroup = ThisWorkbook.Sheets("Label Creator" ).Shapes.Range((GrpArr)).Group
35     With TemplateGroup
36         .Name = "TemplateGroup"
37         .LockAspectRatio = msoCTrue
38         .Placement = xlMove
39         .Visible = msoFalse
40     End With
41 End With
42 NoGroup:
43 End Sub
44
45 Sub Show_LabelPic()
46     Dim PicPath As String
47     With Sheet1
48         On Error Resume Next
49         .Shapes("LabelPic" ).Delete 'Delete Picture if it exists
50         On Error GoTo 0
51         PicPath = .Range("B84" ).Value 'Path of the picture
52         If PicPath = Empty Then Exit Sub
53         With .Pictures.Insert(PicPath)
54             With .ShapeRange
55                 .LockAspectRatio = msoTrue
56                 .Height = 95
57                 .Name = "LabelPic"
58             End With 'Shape Range
59         End With

```

1 2 3

```
60 1 2 3 End With 'Pictures
61
62 With .Shapes("LabelPic" )
63 .Left = Sheet1.Range("H84" ).Left
64 .Top = Sheet1.Range("H84" ).Top
65 .IncrementLeft 20
66 .IncrementTop 15
67 .ZOrder msoSendToBack
68 End With
69 .Shapes("BackLabShape" ).ZOrder msoSendToBack
70 End With
71 End Sub
72
73
74 Sub UngroupTemplateShape()
75 On Error Resume Next
76 With Sheet1.Shapes("TemplateGroup" )
77 .Visible = msoCTrue
78 .Ungroup
79 End With
80 On Error GoTo 0
81 End Sub
```

**A**

Add, 33  
Add\_LabelPic, 33  
Application, 33

**D**

Delete, 33

**E**

Empty, 33  
Explicit, 33

**F**

FileDialog, 33  
Filters, 33

**G**

Group, 33  
GroupTemplateShapes, 33  
GrpArr, 33

**H**

Height, 33

**I**

IncrementLeft, 34  
IncrementTop, 34  
Insert, 33  
InStr, 33

**L**

LabelShape, 33  
Left, 33, 34  
Len, 33  
LockAspectRatio, 33

**M**

msoCTrue, 33, 34  
msoFalse, 33  
msoFileDialogFilePicker, 33  
msoSendToBack, 34  
msoTrue, 33

**N**

Name, 33  
NoGroup, 33  
NoSelection, 33

**P**

PicFile, 33  
PicPath, 33  
Pictures, 33  
Placement, 33

**R**

Range, 33, 34

**S**

SelectedItems, 33  
Shape, 33  
ShapeRange, 33  
Shapes, 33, 34  
ShapesNames, 33  
Sheet1, 33, 34  
Sheets, 33  
Show, 33  
Show\_LabelPic, 33  
Split, 33

**T**

TemplateGroup, 33  
ThisWorkbook, 33  
Title, 33  
Top, 34

**U**

Ungroup, 34  
UngroupTemplateShape, 34

**V**

Value, 33  
Visible, 33, 34

**X**

xLMove, 33

**Z**

ZOrder, 34

```
1 Option Explicit
2
3 Sub CreateLabelStep1()
4 StopCalc 'Import Step 1
5 GroupTemplateShapes
6 ResetHideSteps
7 Sheet1.Shapes("Step1").Fill.ForeColor.RGB = RGB(50, 255, 50)
8 Sheet1.Shapes("Step1").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
9 Sheet1.Range("5:16").EntireRow.Hidden = False
10 Sheet1.Shapes("Step1Grp").Visible = msoCTrue
11 On Error Resume Next
12 Sheet1.Shapes("LabSheetPic").Visible = msoCTrue
13 Sheet1.Range("E8").Select
14 On Error GoTo 0
15 ResetCalc
16 End Sub
17
18 Sub CreateLabelStep2()
19 StopCalc 'Import Step 2
20 GroupTemplateShapes
21 ResetHideSteps
22 Sheet1.Range("17:77").EntireRow.Hidden = False
23 Sheet1.Shapes("Step2").Fill.ForeColor.RGB = RGB(50, 255, 50)
24 Sheet1.Shapes("Step2").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
25 Sheet1.Shapes("Step2Grp").Visible = msoCTrue
26 Sheet1.Range("E20").Select
27 ResetCalc
28 End Sub
29
30 Sub CreateLabelStep3()
31 If Sheet1.Range("E20").Value = Empty Then
32 MsgBox "Please select a data file for your labels before moving onto the next step"
33 Exit Sub
34 End If
35 StopCalc 'Import Step 3
36 ResetHideSteps
37 Sheet1.Range("78:132").EntireRow.Hidden = False
38 Sheet1.Shapes("Step3").Fill.ForeColor.RGB = RGB(50, 255, 50)
39 Sheet1.Shapes("Step3").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
40 Sheet1.Shapes("Step3Grp").Visible = msoCTrue
41 UngroupTemplateShape 'Ungroup any lable shapes (if existing)
42 With Sheet1.Shapes("BackLabShape")
43 .Height = Application.InchesToPoints(Sheet1.Range("E10").Value)
44 .Width = Application.InchesToPoints(Sheet1.Range("G10").Value)
45 .Visible = msoCTrue
46 End With
47 ImportLabelData
48 ResetCalc
49 End Sub
50
51 Sub CreateLabelStep4()
52 StopCalc 'Import Step 4
53 GroupTemplateShapes 'Group Shapes before moving to Step 4
54 ResetHideSteps
55 Sheet1.Range("133:180").EntireRow.Hidden = False
56 Sheet1.Shapes("Step4").Fill.ForeColor.RGB = RGB(50, 255, 50)
57 Sheet1.Shapes("Step4").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
58 Sheet1.Shapes("Step4Grp").Visible = msoCTrue
59 BuildLabelSheet
60 ResetCalc
61 FilterDataCreateLabels
```

1

```
1
62 End Sub
63
64 Sub ResetHideSteps()
65     With Sheet1 'Hide All Shapes & Rows
66         .Range("5:180").EntireRow.Hidden = True
67         .Shapes.Range(Array("ButtonSet")).Fill.ForeColor.RGB = RGB(0, 97, 0)
68         .Shapes.Range(Array("ButtonSet")).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(
255, 255, 255)
69         .Shapes("Step1Grp").Visible = msoFalse
70         .Shapes("Step2Grp").Visible = msoFalse
71         .Shapes("Step3Grp").Visible = msoFalse
72         .Shapes("Step4Grp").Visible = msoFalse
73         On Error Resume Next
74         Sheet1.Shapes("LabSheetPic").Visible = msoFalse 'Hide Label Sheet Icon
75         Sheet1.Shapes("BackLabShape").Visible = msoFalse 'Hide LabelTemplate Background
76         Sheet1.Shapes("LabPreview").Visible = msoFalse
77         On Error GoTo 0
78     End With
79 End Sub
```



## A

Application, 36  
Array, 37

## B

BuildLabelSheet, 36

## C

CreateLabelStep1, 36  
CreateLabelStep2, 36  
CreateLabelStep3, 36  
CreateLabelStep4, 36

## E

Empty, 36  
EntireRow, 36, 37  
Explicit, 36

## F

Fill, 36, 37  
FilterDataCreateLabels, 36  
Font, 36, 37  
ForeColor, 36, 37

## G

GroupTemplateShapes, 36

## H

Height, 36  
Hidden, 36, 37

## I

ImportLabelData, 36  
InchesToPoints, 36

## M

MsgBox, 36  
msoCTrue, 36  
msoFalse, 37

## R

Range, 36, 37  
ResetCalc, 36  
ResetHideSteps, 36, 37  
RGB, 36, 37

## S

Shapes, 36, 37  
Sheet1, 36, 37  
StopCalc, 36

## T

TextFrame2, 36, 37  
TextRange, 36, 37

## U

UngroupTemplateShape, 36

## V

Value, 36  
Visible, 36, 37

## W

Width, 36

# 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*