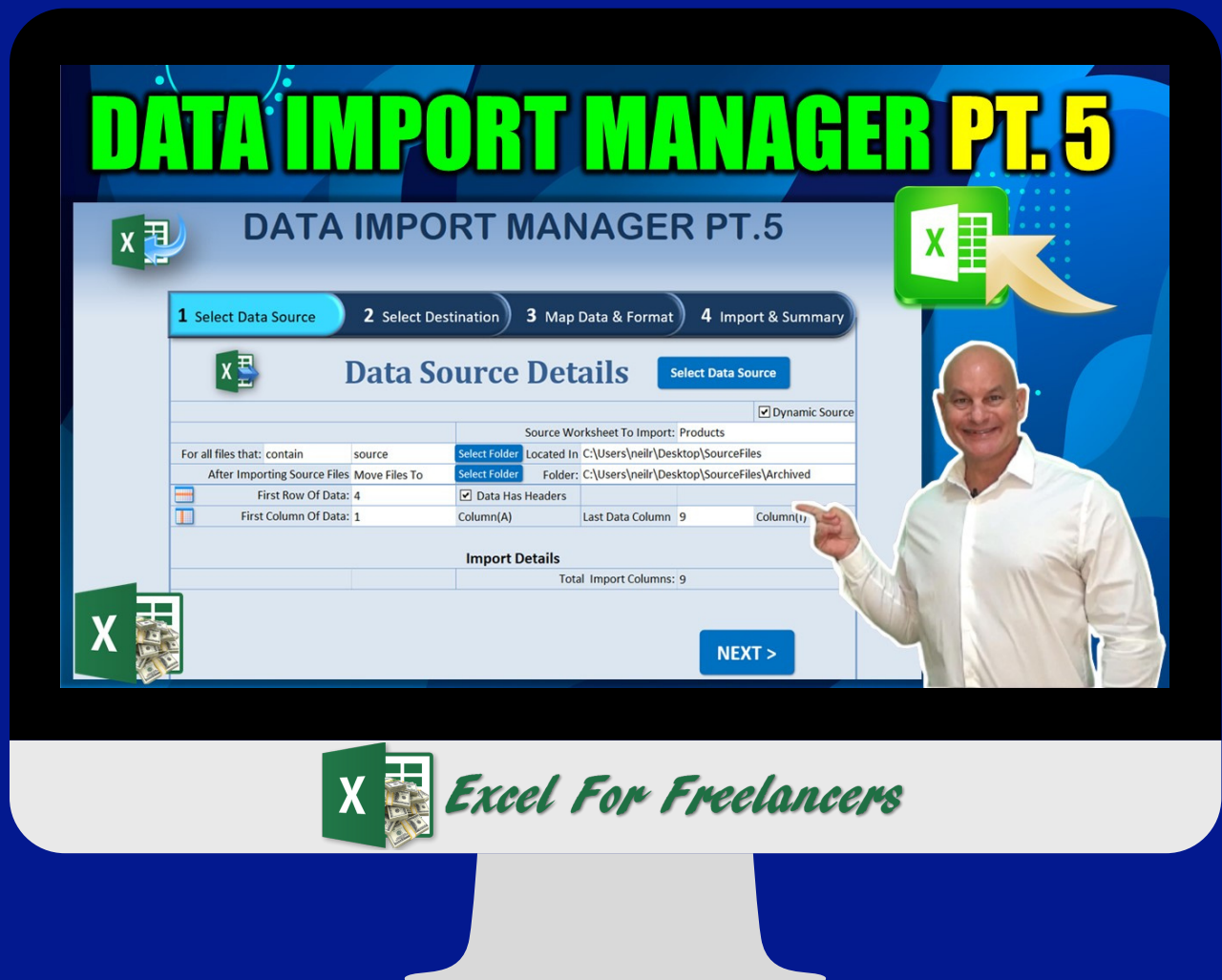


VBA SOURCE CODE BOOK



Excel For Freelancers

**How To Import Multiple Excel
Files Into A Single Destination
In One Click
[Import Mgr. Pt 5]**



**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



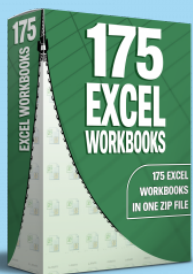
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[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 |
| Sheet2..... | 4 |
| (Declarations)..... | 4 |
| Sheet3..... | 6 |
| (Declarations)..... | 6 |
| Sheet4..... | 8 |
| (Declarations)..... | 8 |
| ThisWorkbook | 10 |
| (Declarations)..... | 10 |
| Modules..... | 12 |
| CodeReset_Macs | 12 |
| (Declarations)..... | 12 |
| ResetCalc [Sub] | 12 |
| StopCalc [Sub] | 12 |
| DataMappingMacs | 14 |
| (Declarations)..... | 14 |
| MapData_Load [Sub] | 14 |
| DestinMacros | 19 |
| (Declarations)..... | 19 |
| BrowseForDestination [Sub]..... | 19 |
| GetDestinData [Sub] | 19 |
| UseExistingWorkbook [Sub]..... | 21 |
| UseNewWorkbook [Sub] | 21 |
| ImpDataDynamicMacros | 23 |
| (Declarations)..... | 23 |
| ImportData_Dynamic [Sub] | 23 |
| ImpDataStaticMacros | 29 |
| (Declarations)..... | 29 |
| CloseImportWkBk [Sub] | 29 |
| ImportData_Static [Sub] | 29 |
| RunImportMacro [Sub] | 31 |
| SourceMacros | 33 |
| (Declarations)..... | 33 |
| BrowseDynamicSourceFolder [Sub]..... | 33 |
| BrowseForSource [Sub] | 33 |
| BrowseMoveFilesToFolder [Sub] | 33 |
| DynamicSource [Sub] | 34 |
| GetSourceData [Sub] | 34 |
| TabMacros | 37 |
| (Declarations)..... | 37 |
| ImportStep1 [Sub] | 37 |
| ImportStep2 [Sub] | 37 |
| ImportStep3 [Sub] | 37 |
| ImportStep4 [Sub] | 37 |
| ResetHideSteps [Sub] | 37 |

Option Explicit

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

```
    If Not Intersect(Target, Range("I9" )) Is Nothing Then GetSourceData 'Get Source workbook  
    data on worksheet Change
```

```
    If Not Intersect(Target, Range("I29" )) Is Nothing Then GetDestinData 'Get  
    Destinationworksheet data on worksheet change
```

```
    '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("H64:H163" )) Is Nothing And Range("D" & Target.Row).Value  
    <> Empty And Range("B62" ).Value = False 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("I64:I163" )) 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
```

```
    'On Change of 'After Importing Files Action' Display/Hide Select Folder button
```

```
    If Not Intersect(Target, Range("F11" )) Is Nothing Then
```

```
        If Target.Value = "Move Files To" Then Shapes("MoveToFolderBtn" ).Visible = msoCTrue  
        Else: Shapes("MoveToFolderBtn" ).Visible = msoFalse
```

```
    End If
```

```
End Sub
```

A

Add, [2](#)
AlertStyle, [2](#)

D

Delete, [2](#)

E

Empty, [2](#)
Explicit, [2](#)

F

Formula1, [2](#)

G

GetDestinData, [2](#)
GetSourceData, [2](#)

I

Intersect, [2](#)

M

msoCTrue, [2](#)
msoFalse, [2](#)

N

NumberFormat, [2](#)

O

Operator, [2](#)

R

Range, [2](#)
Row, [2](#)

S

Shapes, [2](#)

T

Target, [2](#)

V

Validation, [2](#)
Value, [2](#)
Visible, [2](#)

W

Worksheet_Change, [2](#)

X

xlBetween, [2](#)
xlValidAlertStop, [2](#)
xlValidateList, [2](#)

1 Option Explicit

2

E

Explicit, 4

1 Option Explicit

2

E

Explicit, 6

1 Option Explicit

2

E

Explicit, 8

| | |
|---|-----------------|
| 1 | Option Explicit |
| 2 | |

E

Explicit, [10](#)

```
1 Option Explicit
2
3 Sub ResetCalc()
4     With Application
5         .EnableEvents = True
6         .Calculation = xlCalculationAutomatic
7         .ScreenUpdating = True
8     End With
9 End Sub
10
11 Sub StopCalc()
12     With Application
13         .EnableEvents = False
14         .Calculation = xlCalculationManual
15         .ScreenUpdating = False
16     End With
17 End Sub
18
```

A

Application, [12](#)

C

Calculation, [12](#)

E

EnableEvents, [12](#)

Explicit, [12](#)

R

ResetCalc, [12](#)

S

ScreenUpdating, [12](#)

StopCalc, [12](#)

X

xlCalculationAutomatic, [12](#)

xlCalculationManual, [12](#)


```

1  Option Explicit
2
3  Sub MapData_Load()
4      Dim SourceFilePath As String
5      Dim DestinFilePath As String
6      Dim SourceWkBk As Workbook
7      Dim DestinWkBk As Workbook
8      Dim ImportWkBk As Workbook
9      Dim SourceShtName As String
10     Dim SourceFolder As String 'Added
11     Dim SourceFileName As String 'Added
12     Dim DestinShtName As String
13     Dim WsRow As Long
14     Dim DestinHeaderRow As Long
15     Dim TableRow As Long
16     Dim TableCol As Long
17     Dim ColCount As Long
18     Dim FirstSourceDataRow As Long
19     Dim FirstSourceCol As Long
20     Dim FirstDestinDataRow As Long
21     Dim FirstDestinCol As Long
22     Dim LastDestinCol As Long
23     Dim DestinColNumb As Long
24     Dim MapRow As Long
25     Dim LastSourceRow As Long
26     Dim LastSourceCol As Long
27     Dim SourceFolderObj As Object
28     Dim SourceObj As Object
29
30
31     Set ImportWkBk = ThisWorkbook 'Current Import Workbook
32     With Sheet1
33         'Check For Source Workbook
34         If .Range("E8").Value = Empty Then
35             MsgBox "Please select a Source Workbook to import"
36             ImportStep1
37             Exit Sub
38         End If
39
40         'Check For Destination Workbook (if a new workbook was not selected)
41         If .Range("B4").Value = 1 And .Range("E28").Value = Empty Then
42             MsgBox "Please select a Destination Workbook, or choose 'New Workbook' to import"
43             ImportStep2
44             Exit Sub
45         End If
46
47         'Clear Fields, Formats & Validations
48         .Range("B62").Value = True 'Set Mapping Load to True
49         .Range("A65:B165,D65:L165").ClearContents 'Clear Mapping Data
50         .Range("I65:J165").Validation.Delete 'Delete any existing Data Validation
51         .Range("E65:E165").NumberFormat = "General" 'Reset All Formats
52
53
54         'Check for Static or Dynamic Source
55         If .Range("B2").Value = False Then 'Static Source
56             SourceFilePath = .Range("E8").Value 'Source Workbook File Path
57         Else: 'Dynamic Source
58             SourceFileName = .Range("F10").Value 'Source File Name variable
59             'Determine Source File Types
60             Select Case .Range("E10").Value 'File Filter Type
61                 Case Is = "start with"

```

1 2 3 4

```

62   1 2 3 4 SourceFileName = SourceFileName & "*"
63   Case Is = "end with"
64     SourceFileName = "*" & SourceFileName
65   Case Is = "match"
66     SourceFileName = SourceFileName
67   Case Is = "contain"
68     SourceFileName = "*" & SourceFileName & "*"
69   End Select
70
71   SourceFolder = .Range("H10" ).Value
72   Set SourceObj = VBA.CreateObject("Scripting.FileSystemObject" )
73   Set SourceFolderObj = SourceObj.GetFolder(SourceFolder)
74   For Each SourceObj In SourceFolderObj.Files
75     If LCase(SourceObj.Name) Like LCase(SourceFileName) Then
76       Exit For
77     End If
78   Next SourceObj
79
80   On Error Resume Next
81   SourceFilePath = SourceObj.Path 'If no matches are found, an error would occur here
82
83   On Error GoTo 0
84   If SourceFilePath = "" Then 'If Source File Path is empty then no matches were
85     found
86     MsgBox "No source files, based on your matching requirements were found."
87     Exit Sub
88   End If
89
90   On Error Resume Next 'An error would occur if the workbook is not open
91   Set SourceWkBk = Workbooks(Dir(SourceFilePath)) 'Set Source Workbook
92   If SourceWkBk Is Nothing Then 'If Source Workbook Is Nothing that means it's not open,
93     so we need to open it
94     Set SourceWkBk = Workbooks.Open(SourceFilePath)
95     ThisWorkbook.Activate
96   End If
97   On Error GoTo 0
98
99   SourceShtName = .Range("I9" ).Value 'Source Worksheet Name
100
101   If .Range("B4" ).Value = 1 Then 'Existing Destination Workbook
102     DestinFilePath = .Range("E28" ).Value 'Destination Workbook File Path
103     On Error Resume Next
104     Set DestinWkBk = Workbooks(Dir(DestinFilePath)) 'Set Destination Workbook
105     If DestinWkBk Is Nothing Then
106       Set DestinWkBk = Workbooks.Open(DestinFilePath)
107       ThisWorkbook.Activate
108     End If
109     On Error GoTo 0
110   End If
111
112   'Get Source Data Variables
113   FirstSourceDataRow = .Range("F12" ).Value 'First Row of Source Data
114   LastSourceRow = .Range("I12" ).Value 'Last Row Of Source Data
115   FirstSourceCol = .Range("F13" ).Value 'First Column Of Source Data
116   LastSourceCol = .Range("I13" ).Value 'Last Column Of Source Data
117   ColCount = 1 'Set Initial Column Count
118
119   'Get Destination Data Variables
120   DestinShtName = .Range("I29" ).Value 'Destination Sheet Name

```

```

1 2
120 FirstDestinCol = .Range("F32" ).Value
121 LastDestinCol = .Range("J32" ).Value
122 FirstDestinDataRow = .Range("F31" ).Value
123 'If the Destination is a New Workbook, or New Sheet, set the # of Columns to default
to the # of Source Columns, otherwise the # of Columns is set to the Destination
Workbook Columns
124 If .Range("B4" ).Value = 2 Or .Range("I29" ).Value = "<New Worksheet>" Then
DestinColNumb = .Range("I16" ).Value Else: DestinColNumb = .Range("J32" ).Value
125
126
127 'Load Source Table Headers
128 MapRow = 65
129 For TableCol = FirstSourceCol To LastSourceCol
130     If .Range("A6" ).Value = True Then 'Headers Exist
131         .Range("D" & MapRow).Value = SourceWkBk.Sheets(SourceShtName).Cells(
FirstSourceDataRow - 1, TableCol).Value
132     Else: ' No Headers
133         .Range("D" & MapRow).Value = "Column" & ColCount
134         ColCount = ColCount + 1
135     End If
136     .Range("B" & MapRow).Value = TableCol 'Source Column Number
137     MapRow = MapRow + 1
138 Next TableCol
139
140
141 'Load Source Data & Source Format(First Row)
142 MapRow = 65 'Reset Map Row
143 For TableCol = FirstSourceCol To LastSourceCol
144     .Range("E" & MapRow).Value = SourceWkBk.Sheets(SourceShtName).Cells(
FirstSourceDataRow, TableCol).Value
145     .Range("L" & MapRow).Value = "=Cell(" & Chr(34) & "format" & Chr(34) & "," & '[' &
Dir(SourceFilePath) & "]" & SourceShtName & "!" & SourceWkBk.Sheets(SourceShtName
).Cells(FirstSourceDataRow, TableCol).Address & ")"
146     MapRow = MapRow + 1
147 Next TableCol
148
149 'Load Destination Column
150 MapRow = 65 'Reset Map Row
151 If .Range("B4" ).Value = 2 Or .Range("I29" ).Value = "<New Worksheet>" Then 'New
Destination Workbook or New Sheet
152     For TableCol = 1 To DestinColNumb - 1
153         .Range("G" & MapRow).Value = "Column" & FirstDestinCol & "(" & Replace(Replace
(.Cells(1, FirstDestinCol).Address, "1" , "" ), "$" , "" ) & ")" 'Set Default
Column
154         .Range("A" & MapRow).Value = "Column" & FirstDestinCol & "(" & Replace(Replace
(.Cells(1, FirstDestinCol).Address, "1" , "" ), "$" , "" ) & ")" 'Build Drop
Down Destination List
155         .Range("H" & MapRow).Value = .Range("N" & MapRow).Value 'Add In Cell Format
Type
156         If .Range("N" & MapRow).Value <> Empty And .Range("N" & MapRow).Value <>
"General" Then
157             .Range("I" & MapRow).Validation.Add Type:=xlValidateList, AlertStyle:=
xlValidAlertStop, Operator:=xlBetween, Formula1:="=" & .Range("N" & MapRow).
Value & "FormatList"
158             .Range("I" & MapRow).Value = .Range("M" & MapRow).Value 'Default Format
from source
159         End If
160         MapRow = MapRow + 1 'Increment Mappnig Row
161         FirstDestinCol = FirstDestinCol + 1 'Increment One Column To Right
162     Next TableCol
163 Else: 'Existing Table Destination

```

1 2 3

```
164     1 2 3 For TableCol = FirstDestinCol To LastDestinCol
165     .Range("B6").Value = True Then DestinHeaderRow = .Range("F31").Value - 1
166     Else DestinHeaderRow = .Range("F31").Value
167     .Range("G" & MapRow).Value = DestinWkBk.Sheets(DestinShtName).Cells(
DestinHeaderRow, TableCol).Value
167     .Range("A" & MapRow).Value = DestinWkBk.Sheets(DestinShtName).Cells(
DestinHeaderRow, TableCol).Value
168     'Determine Cell Formats
169
170     .Range("H" & MapRow).Value = .Range("N" & MapRow).Value 'Add In Cell Format
Type
171     If .Range("N" & MapRow).Value <> Empty And .Range("N" & MapRow).Value <>
"General" Then 'Update Cell Validation Type if Not general
172     .Range("I" & MapRow).Validation.Add Type:=xlValidateList, AlertStyle:=
xlValidAlertStop, Operator:=xlBetween, Formula1:="=" & .Range("N" & MapRow).
Value & "FormatList"
173     .Range("I" & MapRow).Value = .Range("M" & MapRow).Value 'Default Format
from Source
174     End If
175     MapRow = MapRow + 1 'Increment Mapping Row
176     Next TableCol
177     End If
178
179     .Range("B61").Value = True 'Set Mapping Completed to True
180     .Range("B62").Value = False 'Set Mapping Load to False
181 End With
182 End Sub
```

A

Activate, 15
Add, 16, 17
Address, 16
AlertStyle, 16, 17

C

Cells, 16, 17
Chr, 16
ClearContents, 14
ColCount, 14-16
CreateObject, 15

D

Delete, 14
DestinColNumb, 14, 16
DestinFilePath, 14, 15
DestinHeaderRow, 14, 17
DestinShtName, 14, 15, 17
DestinWkBk, 14, 15, 17
Dir, 15, 16

E

Empty, 14, 16, 17
Explicit, 14

F

Files, 15
FirstDestinCol, 14, 16, 17
FirstDestinDataRow, 14, 16
FirstSourceCol, 14-16
FirstSourceDataRow, 14-16
Formula1, 16, 17

G

GetFolder, 15

I

ImportStep1, 14
ImportStep2, 14
ImportWkBk, 14

L

LastDestinCol, 14, 16, 17
LastSourceCol, 14-16
LastSourceRow, 14, 15
LCase, 15

M

MapData_Load, 14
MapRow, 14, 16, 17
MsgBox, 14, 15

N

Name, 15
NumberFormat, 14

O

Open, 15
Operator, 16, 17

P

Path, 15

R

Range, 14-17
Replace, 16

S

Sheet1, 14
Sheets, 16, 17
SourceFileName, 14, 15
SourceFilePath, 14-16
SourceFolder, 14, 15
SourceFolderObj, 14, 15
SourceObj, 14, 15
SourceShtName, 14-16
SourceWkBk, 14-16

T

TableCol, 14, 16, 17
TableRow, 14
ThisWorkbook, 14, 15

V

Validation, 14, 16, 17
Value, 14-17
VBA, 15

W

Workbook, 14
Workbooks, 15
WsRow, 14

X

xlBetween, 16, 17
xlValidAlertStop, 16, 17
xlValidateList, 16, 17

```

1  Option Explicit
2
3  Sub BrowseForDestination()
4      Dim FileFldr As FileDialog
5      Dim DestinFilePath As String
6      Dim DestinWs As Worksheet
7      Dim ImportWkBk As Workbook
8      Dim DestinWkBk As Workbook
9      Dim WsRow As Long
10     Set ImportWkBk = ThisWorkbook
11     With Sheet1
12         .Range("B8:B100").ClearContents 'Clear any older source worksheets
13         .Range("E47:I53").ClearContents 'Clear Import Summary Message
14         .Range("A65:B165,D65:L165").ClearContents 'Clear Mapping Data"
15         .Range("I65:J165").Validation.Delete 'Delete any existing Data Validation
16         WsRow = 8 'Set Inital Worksheet Row to 8
17         Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
18         With FileFldr
19             .Title = "Select Destination File"
20             .Filters.Add "Excel Files", "*.xlsm,*.xlsx,*.xlsb, *.xls"
21             If .Show <> -1 Then GoTo NoSelection
22             Sheet1.Range("E28").Value = .SelectedItems(1) 'Get Full File Path
23         End With
24
25         DestinFilePath = .Range("E28").Value 'Long File Path
26         .Range("E29").Value = Dir(DestinFilePath, vbDirectory) 'File Name
27         On Error Resume Next
28         Set DestinWkBk = Workbooks(Dir(DestinFilePath)) 'Set Destination Workbook
29         If DestinWkBk Is Nothing Then
30             Set DestinWkBk = Workbooks.Open(DestinFilePath) 'Open Destination Workbook
31         End If
32
33         For Each DestinWs In DestinWkBk.Worksheets
34             Sheet1.Range("B" & WsRow).Value = DestinWs.Name
35             WsRow = WsRow + 1
36         Next
37         .Range("B" & WsRow).Value = "<New Worksheet>"
38         .Range("I29").Value = .Range("B8").Value 'Set Inital Sheet Name
39         NoSelection:
40         ThisWorkbook.Activate
41     End With
42 End Sub
43 Sub GetDestinData()
44     Dim DestinFilePath As String
45     Dim DestinShtName As String
46     Dim DestinWkBk As Workbook
47     Dim ImportWkBk As Workbook
48     Dim WsRow As Long
49     Dim FirstRow As Long
50     Dim TableRow As Long
51     Dim TableCol As Long
52     Dim FirstCol As Long
53     Dim LastRow As Long
54     Dim LastCol As Long
55
56     Set ImportWkBk = ThisWorkbook
57     With Sheet1
58         .Range("A65:B165,D65:L165").ClearContents 'Clear Mapping Data"
59         .Range("I65:J165").Validation.Delete 'Delete any existing Data Validation
60         .Range("E65:E165").NumberFormat = "General" 'Reset All Formats
61         .Range("E47:I53").ClearContents 'Clear Import Summary Message

```

```

62
63 1 2 If .Range("B4").Value = 1 And Range("I29").Value <> "<New Worksheet>" Then
'Existing Workbook (not new)
64 If .Range("E28").Value = Empty Or .Range("E28").Value = "<New Workbook>" Then
Exit Sub 'No workbook selected
65 DestinFilePath = .Range("E28").Value 'Destin Workbook File Path
66 On Error Resume Next
67 Set DestinWkBk = Workbooks(Dir(DestinFilePath)) 'Set Destination Workbook
68 If DestinWkBk Is Nothing Then
69     Set DestinWkBk = Workbooks.Open(DestinFilePath) 'Open Destination Workbook
70 End If
71 DestinShtName = .Range("I29").Value 'Destin Worksheet Name
72
73 With DestinWkBk.Sheets(DestinShtName)
74     'Get Last Row Of Data
75     LastRow = .Cells.Find(What:="*", After:=.Range("A1"), _
76     Lookat:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByRows, SearchDirection:=
xlPrevious, MatchCase:=False).Row
77     'Get Last Column Of Data
78     LastCol = .Cells.Find(What:="*", After:=.Range("A1"), _
79     Lookat:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByColumns, SearchDirection:=
xlPrevious, MatchCase:=False).Column
80     'On Error GoTo 0
81
82     'Determine First Row Of Table w/ 70% of the cells filled
83     For TableRow = 1 To LastRow
84         If (Application.WorksheetFunction.CountA(Range(.Cells(TableRow, 1), .Cells(
TableRow, LastCol))) / LastCol) >= 0.7 Then
85             FirstRow = TableRow
86             Exit For
87         End If
88     Next TableRow
89
90     'Determine First Column Of Table w/ 50% of the cells filled
91     For TableCol = 1 To LastCol
92         If (Application.WorksheetFunction.CountA(Range(.Cells(1, TableCol), .Cells(
LastRow, TableCol))) / LastRow) >= 0.5 Then
93             FirstCol = TableCol
94             Exit For
95         End If
96     Next TableCol
97 End With
98
99 'For existing Destination workbooks with headers, add 1 to First Row
100 If .Range("B6").Value = True And .Range("B4").Value = 1 Then FirstRow = FirstRow +
1
101
102 .Range("F31").Value = FirstRow 'Set First Row Of Destin Data
103 .Range("F33").Value = LastRow 'Set Last Row Of Destin Data
104 .Range("F32").Value = FirstCol 'Set First Column Of Destin Data
105 .Range("J32").Value = LastCol 'Set Last Column Of Destin Destin Data
106
107 Else: 'New Destination Workbook or New Worksheet
108
109 .Range("F31").Value = 2 'Set First Row Of Destin Data
110 .Range("F33").Value = 1 'Set Last Row Of Destin Data
111 .Range("F32").Value = 1 'Set First Column Of Destin Data
112 .Range("J32").Value = 1 'Set Last Column Of Destin Destin Data
113 If .Range("I29").Value = "<New Worksheet>" Then .Range("I30").Value = "New
Sheet" 'Set Default New Worksheet Name
114
1 2 3

```

```
1 2 3
115     End If
116     .Range("B61").Value = False 'Reset Mapping Complete to False
117 End With
118
119     ImportWkBk.Activate
120     Exit Sub
121     MissingDestin:
122     MsgBox "Please check for a correct Destination Workbook"
123 End Sub
124
125 Sub UseExistingWorkbook()
126     With Sheet1
127         .Shapes("DestWkbkBtn").Visible = msoCTrue
128         .Range("E28:J28,I29:J29,B8:B60").ClearContents
129         .Range("A65:B165,D65:L165").ClearContents 'Clear Mapping Data
130         .Range("I65:J165").Validation.Delete 'Delete any existing Data Validation
131         .Range("B61").Value = False 'Reset Mapping to 'Incomplete'
132         .Range("E65:E165").NumberFormat = "General" 'Reset All Formats
133     End With
134 End Sub
135
136 Sub UseNewWorkbook()
137     With Sheet1
138         .Range("E29:F29,B8:B60").ClearContents
139         .Range("A65:B165,D65:L165").ClearContents 'Clear Mapping Data
140         .Range("I65:J165").Validation.Delete 'Delete any existing Data Validation
141         .Range("E65:E165").NumberFormat = "General" 'Reset All Formats
142         .Range("B8, I29").Value = "Sheet1"
143         .Range("B9").Value = "Sheet2"
144         .Range("B10").Value = "Sheet3"
145         .Range("E28").Value = "<New Workbook>"
146         .Range("F31").Value = 2 'Set First Row of Data
147         .Range("F32").Value = 1 'Set First Column of Data
148         .Range("B61").Value = False 'Reset Mapping to 'Incomplete'
149         .Shapes("DestWkbkBtn").Visible = msoFalse
150     End With
151 End Sub
```


→, 20

A

Activate, 19, 21
Add, 19
After, 20
Application, 19, 20

B

BrowseForDestination, 19

C

Cells, 20
ClearContents, 19, 21
Column, 20
CountA, 20

D

Delete, 19, 21
DestinFilePath, 19, 20
DestinShtName, 19, 20
DestinWkBk, 19, 20
DestinWs, 19
Dir, 19, 20

E

Empty, 20
Explicit, 19

F

FileDialog, 19
FileFldr, 19
Filters, 19
Find, 20
FirstCol, 19, 20
FirstRow, 19, 20

G

GetDestinData, 19

I

ImportWkBk, 19, 21

L

LastCol, 19, 20
LastRow, 19, 20
Lookat, 20
LookIn, 20

M

MatchCase, 20
MissingDestin, 21
MsgBox, 21
msoCTrue, 21
msoFalse, 21
msoFileDialogFilePicker, 19

N

Name, 19
NoSelection, 19
NumberFormat, 19, 21

O

Open, 19, 20

R

Range, 19-21
Row, 20

S

SearchDirection, 20
SearchOrder, 20
SelectedItems, 19
Shapes, 21
Sheet1, 19, 21
Sheets, 20
Show, 19

T

TableCol, 19, 20
TableRow, 19, 20
ThisWorkbook, 19
Title, 19

U

UseExistingWorkbook, 21
UseNewWorkbook, 21

V

Validation, 19, 21
Value, 19-21
vbDirectory, 19
Visible, 21

W

What, 20
Workbook, 19
Workbooks, 19, 20
Worksheet, 19
WorksheetFunction, 20
Worksheets, 19
WsRow, 19

X

xlByColumns, 20
xlByRows, 20
xlFormulas, 20
xlPart, 20
xlPrevious, 20

Option Explicit

Sub ImportData_Dynamic()

Dim SourceFilePath As String

Dim DestinFilePath As String

Dim SourceWkBk As Workbook

Dim DestinWkBk As Workbook

Dim ImportWkBk As Workbook

Dim NewDestinSht As Worksheet

Dim SourceShtName As String

Dim DestinShtName As String

Dim SourceFile As String 'Added

Dim SourceFileName As String 'Added

Dim SourceFolder As String 'Added

Dim MoveSourceToFolder As String 'Added

Dim SourceObj As Object 'Added

Dim SourceFolderObj As Object 'Added

Dim FirstSourceDataRow As Long

Dim FirstSourceCol As Long

Dim LastSourceCol As Long

Dim LastSourceRow As Long

Dim SourceCol As Long

Dim FirstDestinDataRow As Long

Dim FirstDestinCol As Long

Dim LastDestinCol As Long

Dim LastDestinDataRow As Long

Dim DestinCol As Long

Dim TotalImportRows As Long

Dim TotalCombinedRows As Long 'Added

Dim SourceWkBkCount As Long 'Added

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 Data Mapping has been done

If .Range("B61").Value = False Then 'Mapping incomplete

MsgBox "Please complete Data Mapping before import"

ImportStep3

Exit Sub

End If

'Check For Required Dynamic Source Fields

If .Range("E10").Value = "" Or .Range("F10").Value = "" Or .Range("H10").Value = "" Or .Range("I9").Value = "" Then

MsgBox "Please make sure all dynamic source fields have been added"

ImportStep1

Exit Sub

End If

```

61
62 Set ImportWkBk = ThisWorkbook 'Set Current Workbook To Import
63
64 'Get Destination Workbook Variables
65 If .Range("B4").Value = 1 Then 'Use Existing Workbook
66     DestinFilePath = .Range("E28").Value 'Destination file Pat
67     On Error Resume Next 'If workbook is not open, a but would display
68     Set DestinWkBk = Workbooks(Dir(DestinFilePath)) 'Set Destination Workbook
69     If DestinWkBk Is Nothing Then 'Workbook is not open, so open it
70         Set DestinWkBk = Workbooks.Open(DestinFilePath)
71     End If
72     LastDestinDataRow = .Range("F33").Value 'Last Row of Data in Destination
73 Else: 'Use New Workbook
74     Set DestinWkBk = Workbooks.Add 'Set New Destination Workbook
75     DestinFilePath = .Range("H10").Value & "\" & .Range("E29").Value & ".xlsx" 'Set
76     Destination File Path To Source Location To Save New Workbook
77 End If
78 DestinShtName = .Range("I29").Value 'Destination Worksheet Name
79 FirstDestinDataRow = .Range("F31").Value 'First Row of Data in Destination
80 FirstDestinCol = .Range("F32").Value 'First Column of Destination Data
81 LastDestinCol = .Range("J32").Value 'Last Column Of Destination Data
82
83 'Create A New Worksheet on Existing Workbook and '<New Worksheet>' Chose
84 If .Range("B4").Value = 1 And DestinShtName = "<New Worksheet>" Then
85     On Error GoTo DuplicateSheet
86     Set NewDestinSht = DestinWkBk.Sheets.Add(After:=DestinWkBk.Sheets(DestinWkBk.Sheets.
87     Count))
88     If .Range("I30").Value <> Empty Then NewDestinSht.Name = .Range("I30").Value Else:
89         NewDestinSht.Name = "New Sheet" 'Set New Sheet Name to Default or user set
90     DestinShtName = NewDestinSht.Name 'Rename Variable to newly created sheet name
91 End If
92
93 'Get First Available Row In Destination
94 If .Range("B4").Value = 1 And .Range("I29").Value <> "<New Worksheet>" And .Range(
95     "I31").Value = "Add To Existing Data" Then 'Add To Existing Data (Do Not Replace)
96     FirstAvailRow = LastDestinDataRow + 1 'Last Row In Destination With a value + 1
97 Else:
98     FirstAvailRow = FirstDestinDataRow
99 End If
100
101 'Get Dynamic Source Workbook Details
102
103 'Get Source Workbook Variables
104 FirstSourceDataRow = .Range("F12").Value 'First Row of Data (excludes headers)
105 FirstSourceCol = .Range("F13").Value 'First Column Of Data from the source sheet
106 LastSourceCol = .Range("I13").Value 'Last Source Column
107 SourceShtName = .Range("I9").Value 'Source Worksheet Name
108 SourceFolder = .Range("H10").Value 'Source Filder
109 SourceFileName = .Range("F10").Value 'Initial Source File Name
110 MoveSourceToFolder = .Range("H11").Value 'Move To Folder (If applicaple)
111
112 'Determine Source File Types
113 Select Case .Range("E10").Value 'File Filter Type
114 Case Is = "start with"
115     SourceFileName = SourceFileName & "*"
116 Case Is = "end with"
117     SourceFileName = "*" & SourceFileName
118 Case Is = "match"
119     SourceFileName = SourceFileName
120 Case Is = "contain"
121     SourceFileName = "*" & SourceFileName & "*"

```

```

118 End Select
119
120 Set SourceObj = VBA.CreateObject("Scripting.FileSystemObject" )
121 Set SourceFolderObj = SourceObj.GetFolder(SourceFolder)
122
123 'Begin Loop For All Source Workbooks
124 For Each SourceObj In SourceFolderObj.Files
125     If LCase(SourceObj.Name) Like LCase(SourceFileName) Then 'Check For File match
126         SourceFilePath = SourceObj.Path
127         SourceFile = Dir(SourceFilePath)
128         SourceWkBkCount = SourceWkBkCount + 1 'Count # of workbooks being Imported
129         On Error Resume Next 'An error will occur if we set a workbook that is not open.
130         'If the error occurs we can run a check and open the workbook
131         Set SourceWkBk = Workbooks(SourceFile) 'Set Source Workbook
132         If SourceWkBk Is Nothing Then 'Source Workbook is not open
133             Set SourceWkBk = Workbooks.Open(SourceFilePath)
134         End If
135         LastSourceRow = SourceWkBk.Sheets(SourceShtName).Cells.Find(What:="*", After:=.
136         Range("A1" ), Lookat:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByRows,
137         SearchDirection:=xlPrevious, MatchCase:=False).Row 'Last Row Of Data
138         TotalImportRows = LastSourceRow - FirstSourceDataRow + 1 'Total Rows To Be
139         Imported
140
141         'Copy Over Data & Formats From Source To Destination
142         LastMapCol = Application.WorksheetFunction.CountA(.Range("A65:A163" )) + 64 'Use
143         instad of End(xlup) for hidden rows & columns
144
145         TableCol = FirstDestinCol 'Set Intial Table Column
146         'Check to See if we are copying over headers and only on the first of many
147         source workbooks
148         If SourceWkBkCount = 1 And .Range("B4" ).Value = 2 Or .Range("I29" ).Value =
149         "<New Worksheet>" Then 'New Workbook or New Worksheet, add headers
150             For MapCol = 65 To LastMapCol
151                 If .Range("G" & MapCol).Value <> "<Do Not Import>" Then
152                     DestinCol = .Range("O" & MapCol).Value + FirstDestinCol - 1
153                     DestinWkBk.Sheets(DestinShtName).Cells(FirstDestinDataRow - 1, DestinCol
154                     ).Value = .Range("D" & MapCol).Value 'Add Column Header
155                     TableCol = TableCol + 1 'Increase The Column Count By 1
156                 End If
157             Next MapCol
158         End If
159
160         'If Using Existing Destination Workbook, On an Exsting Sheet AND Replacing Data,
161         then clear old data
162         If .Range("B4" ).Value = 1 And .Range("I29" ).Value <> "<New Worksheet>" And .
163         Range("I31" ).Value = "Replace Exisitng Data" Then
164             Range(DestinWkBk.Sheets(DestinShtName).Cells(FirstDestinDataRow,
165             FirstDestinCol), DestinWkBk.Sheets(DestinShtName).Cells(LastDestinDataRow,
166             LastDestinCol)).ClearContents
167         End If
168
169         'Copy Over Data & Formats
170         TableCol = FirstDestinCol 'Reset Intial Table Column
171         For MapCol = 65 To LastMapCol 'For Each Column in Source From Data Mapping
172             If .Range("G" & MapCol).Value <> "<Do Not Import>" Then
173                 SourceCol = .Range("B" & MapCol).Value 'Determine Source Column
174                 DestinCol = .Range("O" & MapCol).Value + FirstDestinCol - 1 'Determine
175                 Destination Column
176                 Range(DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow, DestinCol),
177                 DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow + TotalImportRows - 1,

```

```

1 2 3 4 5 6
165     DestinCol)).Value = _
166     Range(SourceWkBk.Sheets(SourceShtName).Cells(FirstSourceDataRow, SourceCol)
167     , SourceWkBk.Sheets(SourceShtName).Cells(LastSourceRow, SourceCol)).Value
168     'Add In Formats
169     If .Range("I" & MapCol).Value <> "" Then CellFormat = .Range("I" &
170     MapCol).Value Else CellFormat = "General" 'Determine Cell Format
171     Range(DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow, DestinCol),
172     DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow + TotalImportRows - 1,
173     DestinCol)).NumberFormat = CellFormat
174     TableCol = TableCol + 1 'Create a column count and increase it by 1
175     End If
176     Next MapCol
177
178     'Close Source Workbook
179     SourceWkBk.Close False
180
181     'Move Source Workbook if requested
182     If .Range("F11").Value = "Move Files To" And MoveSourceToFolder <> Empty Then
183     Name SourceFilePath As MoveSourceToFolder & "/" & Dir(SourceFilePath)
184     End If
185
186     'Delete File if option was requested
187     If .Range("F11").Value = "Delete Files" Then
188     Kill (SourceFilePath)
189     End If
190
191     Set SourceWkBk = Nothing
192     FirstAvailRow = FirstAvailRow + TotalImportRows 'Update First Available Row For
193     Next Import
194     TotalCombinedRows = TotalCombinedRows + TotalImportRows 'Update Total Combined
195     Rows
196     End If 'File Match
197     Next SourceObj 'Next Matching Workbook In Folder
198
199     If SourceWkBkCount <> 0 Then 'Source Workbooks Found
200     Range(DestinWkBk.Sheets(DestinShtName).Cells(1, FirstDestinCol), DestinWkBk.Sheets(
201     DestinShtName).Cells(1, LastDestinCol)).EntireColumn.AutoFit 'Autofit all columns
202     in destination
203
204     'Save Destination Workbook (if existing), or Save As (if new)
205     If .Range("B4").Value = 1 Then
206     DestinWkBk.Save
207     Else
208     If Dir(DestinFilePath) <> "" Then Kill (DestinFilePath) 'Delete if existing to
209     prevent errors
210     DestinWkBk.SaveAs DestinFilePath
211     End If
212
213     'Completion Message
214     .Range("E47").Value = "Your import has successfully been completed." & vbCrLf &
215     "A total of " & TotalCombinedRows & " rows have been imported to " & Dir(
216     DestinFilePath) & " on Sheet " & DestinShtName
217     Else: 'No Source Workbooks Found
218     .Range("E47").Value = "No Source Workbooks have been found"
219     End If
220
221     ImportWkBk.Activate
222     Exit Sub
223 DuplicateSheet:

```

```
214     1 2 MsgBox "The Sheet Name '" & .Range("I30").Value & "' already exists" & vbCrLf &  
215         "Please select a new worksheet name to add"  
216     End With  
217 End Sub  
218  
219  
220
```

—, 26

A

Activate, 26
 Add, 24
 After, 24, 25
 Application, 25
 AutoFit, 26

C

CellFormat, 23, 26
 Cells, 25, 26
 ClearContents, 25
 Close, 26
 ColCount, 23
 Count, 24
 CountA, 25
 CreateObject, 25

D

DestinCol, 23, 25, 26
 DestinFilePath, 23, 24, 26
 DestinShtName, 23-26
 DestinWkBk, 23-26
 Dir, 24-26
 DuplicateSheet, 24, 26

E

Empty, 24, 26
 EntireColumn, 26
 Explicit, 23

F

Files, 25
 Find, 25
 FirstAvailRow, 23-26
 FirstDestinCol, 23-26
 FirstDestinDataRow, 23-25
 FirstSourceCol, 23, 24
 FirstSourceDataRow, 23-26

G

GetFolder, 25

I

ImportData_Dynamic, 23
 ImportStep1, 23
 ImportStep3, 23
 ImportWkBk, 23, 24, 26

K

Kill, 26

L

LastDestinCol, 23-26
 LastDestinDataRow, 23-25
 LastMapCol, 23, 25
 LastSourceCol, 23, 24
 LastSourceRow, 23, 25, 26
 LCase, 25
 Lookat, 25
 LookIn, 25

M

MapCol, 23, 25, 26
 MatchCase, 25
 MoveSourceToFolder, 23, 24, 26
 MsgBox, 23, 27

N

Name, 24-26
 NewDestinSht, 23, 24
 NumberFormat, 26

O

Open, 24, 25

P

Path, 25

R

Range, 23-27
 Row, 25

S

Save, 26
 SaveAs, 26
 SearchDirection, 25
 SearchOrder, 25
 Sheet1, 23
 Sheets, 24-26
 SourceCol, 23, 25, 26
 SourceFile, 23, 25
 SourceFileName, 23-25
 SourceFilePath, 23, 25, 26
 SourceFolder, 23-25
 SourceFolderObj, 23, 25
 SourceObj, 23, 25, 26
 SourceShtName, 23-26
 SourceWkBk, 23, 25, 26
 SourceWkBkCount, 23, 25, 26

T

TableCol, 23, 25, 26
 TableRow, 23
 ThisWorkbook, 24
 TotalCombinedRows, 23, 26
 TotalImportRows, 23, 25, 26

V

Value, 23-27
 VBA, 25
 vbCrLf, 26, 27

W

What, 25
 Workbook, 23
 Workbooks, 24, 25
 Worksheet, 23
 WorksheetFunction, 25

X

xlByRows, 25
 xlFormulas, 25
 xlPart, 25
 xlPrevious, 25

```
1 Option Explicit
2
3
4 Sub CloseImportWkBk()
5     Sheet1.Range("E47:I53").ClearContents
6     ThisWorkbook.Close True
7 End Sub
8
9
10
11 Sub ImportData_Static()
12     Dim SourceFilePath As String
13     Dim DestinFilePath As String
14
15     Dim SourceWkBk As Workbook
16     Dim DestinWkBk As Workbook
17     Dim ImportWkBk As Workbook
18
19     Dim NewDestinSht As Worksheet
20     Dim SourceShtName As String
21     Dim DestinShtName As String
22
23     Dim FirstSourceDataRow As Long
24     Dim FirstSourceCol As Long
25     Dim LastSourceCol As Long
26     Dim LastSourceRow As Long
27     Dim SourceCol As Long
28
29     Dim FirstDestinDataRow As Long
30     Dim FirstDestinCol As Long
31     Dim LastDestinCol As Long
32     Dim LastDestinDataRow As Long
33     Dim DestinCol As Long
34
35     Dim TotalImportRows As Long
36     Dim TableRow As Long
37     Dim TableCol As Long
38     Dim ColCount As Long
39     Dim FirstAvailRow As Long
40
41
42     Dim MapCol As Long
43     Dim LastMapCol As Long
44     Dim CellFormat As String
45
46     With Sheet1
47         'Make sure Data Mapping has been done
48         If .Range("B61").Value = False Then 'Mapping incomplete
49             MsgBox "Please complete Data Mapping before import"
50             ImportStep3
51             Exit Sub
52         End If
53
54         Set ImportWkBk = ThisWorkbook 'Set Current Workbook To Import
55
56         'Get Source Workbook Variables
57         SourceFilePath = .Range("E8").Value 'Source Workbook File Path
58         On Error Resume Next 'An error will occur if we set a workbook that is not open. If
59         the error occurs we can run a check and open the workbook
60         Set SourceWkBk = Workbooks(Dir(SourceFilePath)) 'Set Source Workbook
61         If SourceWkBk Is Nothing Then 'Source Workbook is not open
```

1 2 3


```

1 2 3
61 Set SourceWkBk = Workbooks.Open(SourceFilePath)
62 End If
63 SourceShtName = .Range("I9").Value 'Source Worksheet Name
64 FirstSourceDataRow = .Range("F12").Value 'First Row of Data (excludes headers)
65 FirstSourceCol = .Range("F13").Value 'First Column Of Data from the source sheet
66 LastSourceCol = .Range("I13").Value 'Last Source Column
67 LastSourceRow = .Range("I12").Value 'Last Row Of Data
68 TotalImportRows = .Range("F16").Value 'Total Rows To Be Imported
69
70
71 'Get Destination Workbook Variables
72 If .Range("B4").Value = 1 Then 'Use Existing Workbook
73     DestinFilePath = .Range("E28").Value 'Destination file Path
74     On Error Resume Next 'If workbook is not open, a but would display
75     Set DestinWkBk = Workbooks(Dir(DestinFilePath)) 'Set Destination Workbook
76     If DestinWkBk Is Nothing Then 'Workbook is not open, so open it
77         Set DestinWkBk = Workbooks.Open(DestinFilePath)
78     End If
79     LastDestinDataRow = .Range("F33").Value 'Last Row of Data in Destination
80 Else: 'Use New Workbook
81     Set DestinWkBk = Workbooks.Add 'Set New Destination Workbook
82     DestinFilePath = SourceWkBk.Path & "\ " & .Range("E29").Value & ".xlsx" 'Set
83     Destination File Path To Save New Workbook
84 End If
85 DestinShtName = .Range("I29").Value 'Destination Worksheet Name
86 FirstDestinDataRow = .Range("F31").Value 'First Row of Data in Destination
87 FirstDestinCol = .Range("F32").Value 'First Column of Destination Data
88 LastDestinCol = .Range("J32").Value 'Last Column Of Destination Data
89
90 'Create A New Worksheet on Existing Workbook if '<New Worksheet>' Chosen
91 If .Range("B4").Value = 1 And DestinShtName = "<New Worksheet>" Then
92     On Error GoTo DuplicateSheet
93     Set NewDestinSht = DestinWkBk.Sheets.Add(After:=DestinWkBk.Sheets(DestinWkBk.Sheets.
94     Count))
95     If .Range("I30").Value <> Empty Then NewDestinSht.Name = .Range("I30").Value Else:
96     NewDestinSht.Name = "New Sheet" 'Set New Sheet Name to Default or user set
97     DestinShtName = NewDestinSht.Name 'Rename Variable to newly created sheet name
98 End If
99
100 'Get First Available Row In Destination
101 If .Range("B4").Value = 1 And .Range("I29").Value <> "<New Worksheet>" And .Range(
102 "I31").Value = "Add To Existing Data" Then 'Add To Existing Data (Do Not Replace)
103     FirstAvailRow = LastDestinDataRow + 1 'Last Row In Destination With a value + 1
104 Else:
105     FirstAvailRow = FirstDestinDataRow
106 End If
107
108 'Copy Over Data & Formats From Source To Destination
109 LastMapCol = Application.WorksheetFunction.CountA(.Range("A65:A163")) + 64 'Use
110 instad of End(xlup) for hidden rows & columns
111
112 TableCol = FirstDestinCol 'Set Intial Table Column
113 'Check to See if we are copying over headers
114 If .Range("B4").Value = 2 Or .Range("I29").Value = "<New Worksheet>" Then 'New
115 Workbook or New Worksheet, add headers
116 For MapCol = 65 To LastMapCol
117     If .Range("G" & MapCol).Value <> "<Do Not Import>" Then
118         DestinCol = .Range("O" & MapCol).Value + FirstDestinCol - 1

```

1 2 3 4 5

```

1 2 3 4 5
116 DestInWkBk.Sheets(DestinShtName).Cells(FirstDestinDataRow - 1, DestinCol).
      Value = .Range("D" & MapCol).Value 'Add Column Header
117 TableCol = TableCol + 1 'Increase The Column Count By 1
118 End If
119 Next MapCol
120 End If
121
122 'If Using Existing Destination Workbook, On an Existing Sheet AND Replacing Data, then
      clear old data
123 If .Range("B4").Value = 1 And .Range("I29").Value <> "<New Worksheet>" And .Range(
      "I31").Value = "Replace Existing Data" Then
124 Range(DestinWkBk.Sheets(DestinShtName).Cells(FirstDestinDataRow, FirstDestinCol),
      DestinWkBk.Sheets(DestinShtName).Cells(LastDestinDataRow, LastDestinCol)).
      ClearContents
125 End If
126
127 'Copy Over Data & Formats
128 TableCol = FirstDestinCol 'Reset Initial Table Column
129 For MapCol = 65 To LastMapCol 'For Each Column in Source From Data Mapping
130 If .Range("G" & MapCol).Value <> "<Do Not Import>" Then
131 SourceCol = .Range("B" & MapCol).Value 'Determine Source Column
132 DestinCol = .Range("O" & MapCol).Value + FirstDestinCol - 1 'Determine
      Destination Column
133 Range(DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow, DestinCol),
      DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow + TotalImportRows - 1,
      DestinCol)).Value = _
134 Range(SourceWkBk.Sheets(SourceShtName).Cells(FirstSourceDataRow, SourceCol),
      SourceWkBk.Sheets(SourceShtName).Cells(LastSourceRow, SourceCol)).Value
135
136 'Add In Formats
137 If .Range("I" & MapCol).Value <> "" Then CellFormat = .Range("I" & MapCol).
      Value Else CellFormat = "General" 'Determine Cell Format
138 Range(DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow, DestinCol),
      DestinWkBk.Sheets(DestinShtName).Cells(FirstAvailRow + TotalImportRows - 1,
      DestinCol)).NumberFormat = CellFormat
139 TableCol = TableCol + 1 'Create a column count and increase it by 1
140 End If
141 Next MapCol
142 Range(DestinWkBk.Sheets(DestinShtName).Cells(1, FirstDestinCol), DestinWkBk.Sheets(
      DestinShtName).Cells(1, DestinCol)).EntireColumn.AutoFit
143 If .Range("B4").Value = 1 Then DestinWkBk.Save Else DestinWkBk.SaveAs DestinFilePath
      'Save Destination Workbook (if existing), or Save As (if new)
144
145 .Range("E47").Value = "Your import has successfully been completed." & vbCrLf & "A
      total of " & TotalImportRows & " rows have been imported to " & DestinWkBk.Name & "
      on Sheet " & DestinShtName
146 ImportWkBk.Activate
147 Exit Sub
148 DuplicateSheet:
149 MsgBox "The Sheet Name '" & .Range("I30").Value & "' already exists" & vbCrLf &
      "Please select a new worksheet name to add"
150 End With
151 End Sub
152
153 Sub RunImportMacro()
154 If Sheet1.Range("B2").Value = True Then ImportData_Dynamic Else: ImportData_Static
155 End Sub

```

→, 31

A

Activate, 31
Add, 30
After, 30
Application, 30
AutoFit, 31

C

CellFormat, 29, 31
Cells, 31
ClearContents, 29, 31
Close, 29
CloseImportWkBk, 29
ColCount, 29
Count, 30
CountA, 30

D

DestinCol, 29-31
DestinFilePath, 29-31
DestinShtName, 29-31
DestinWkBk, 29-31
Dir, 29, 30
DuplicateSheet, 30, 31

E

Empty, 30
EntireColumn, 31
Explicit, 29

F

FirstAvailRow, 29-31
FirstDestinCol, 29-31
FirstDestinDataRow, 29-31
FirstSourceCol, 29, 30
FirstSourceDataRow, 29-31

I

ImportData_Dynamic, 31
ImportData_Static, 29, 31
ImportStep3, 29
ImportWkBk, 29, 31

L

LastDestinCol, 29-31
LastDestinDataRow, 29-31
LastMapCol, 29-31
LastSourceCol, 29, 30
LastSourceRow, 29-31

M

MapCol, 29-31
MsgBox, 29, 31

N

Name, 30, 31
NewDestinSht, 29, 30
NumberFormat, 31

O

Open, 30

P

Path, 30

R

Range, 29-31
RunImportMacro, 31

S

Save, 31
SaveAs, 31
Sheet1, 29, 31
Sheets, 30, 31
SourceCol, 29, 31
SourceFilePath, 29, 30
SourceShtName, 29-31
SourceWkBk, 29-31

T

TableCol, 29-31
TableRow, 29
ThisWorkbook, 29
TotalImportRows, 29-31

V

Value, 29-31
vbCrLf, 31

W

Workbook, 29
Workbooks, 29, 30
Worksheet, 29
WorksheetFunction, 30

```

1  Option Explicit
2
3  Sub BrowseDynamicSourceFolder()
4      Dim DynSourceFold As FileDialog
5      Set DynSourceFold = Application.FileDialog(msoFileDialogFolderPicker)
6      With DynSourceFold
7          .Title = "Select A Source Folder"
8          If .Show <> -1 Then GoTo NoSelection
9          Sheet1.Range("H10" ).Value = .SelectedItems(1) 'Get Full FolderPath
10     End With
11     NoSelection:
12 End Sub
13
14 Sub BrowseForSource()
15     Dim FileFldr As FileDialog
16     Dim SourceFilePath As String
17     Dim SourceWs As Worksheet
18     Dim SourceWkBk As Workbook
19     Dim WsRow As Long
20
21     With Sheet1
22         .Range("A8:A60,F10,H10:J10" ).ClearContents 'Clear any older source worksheets
23         .Range("E28:J28,I29:J29,B8:B60" ).ClearContents
24         .Range("A65:B165,D65:L165" ).ClearContents 'Clear Mapping Data
25         .Range("E47:I53" ).ClearContents 'Clear Import Summary Message
26         .Range("I65:J165" ).Validation.Delete 'Delete any existing Data Validation
27         .Range("B61" ).Value = False 'Reset Mapping to 'Incomplete'
28         WsRow = 8 'Set Inital Worksheet Row to 8
29         Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
30         With FileFldr
31             .Title = "Select Source File"
32             .Filters.Add "Excel Files" , "*.xlsm,*.xlsx,*.xlsb,*.xls"
33             If .Show <> -1 Then GoTo NoSelection
34             Sheet1.Range("E8" ).Value = .SelectedItems(1) 'Get Full File Path
35         End With
36
37         SourceFilePath = .Range("E8" ).Value 'Long File Path
38         .Range("E9" ).Value = Dir(SourceFilePath, vbDirectory) 'File Name
39         Workbooks.Open (SourceFilePath) 'Open Source Workbook
40         Set SourceWkBk = Workbooks(Dir(SourceFilePath)) 'Set Source Workbook
41
42         For Each SourceWs In SourceWkBk.Worksheets
43             Sheet1.Range("A" & WsRow).Value = SourceWs.Name
44             WsRow = WsRow + 1
45         Next
46         .Range("I9" ).Value = .Range("A8" ).Value 'Set Inital Sheet Name
47         NoSelection:
48     End With
49 End Sub
50
51 Sub BrowseMoveFilesToFolder()
52     Dim DynSourceFold As FileDialog
53     Set DynSourceFold = Application.FileDialog(msoFileDialogFolderPicker)
54     With DynSourceFold
55         .Title = "Select A Folder To Move Files After Import  "
56         If .Show <> -1 Then GoTo NoSelection
57         Sheet1.Range("H11" ).Value = .SelectedItems(1) 'Get Full FolderPath
58     End With
59     NoSelection:
60 End Sub
61

```

```

62
63
64
65 Sub DynamicSource()
66     With Sheet1
67         If .Range("B2").Value = False Then 'Static Source
68             .Shapes("SelectSourceBtn").Visible = msoCTrue
69             .Shapes("SelectFolderBtn").Visible = msoFalse
70             .Shapes("MoveToFolderBtn").Visible = msoFalse
71         Else: ' Dynamic Source
72             .Shapes("SelectSourceBtn").Visible = msoFalse
73             .Shapes("SelectFolderBtn").Visible = msoCTrue
74             'Display/Hide Move To Folder Button based on user selection
75             If .Range("F11").Value = "Move Files To" Then .Shapes("MoveToFolderBtn").Visible
76                 = msoCTrue Else .Shapes("MoveToFolderBtn").Visible = msoFalse
77         End If
78     End With
79 End Sub
80
81 Sub GetSourceData()
82     Dim SourceFilePath As String
83     Dim SourceShtName As String
84     Dim SourceWkBk As Workbook
85     Dim ImportWkBk As Workbook
86     Dim WsRow As Long
87     Dim FirstRow As Long
88     Dim TableRow As Long
89     Dim TableCol As Long
90     Dim FirstCol As Long
91     Dim LastRow As Long
92     Dim LastCol As Long
93
94     Set ImportWkBk = ThisWorkbook
95     With Sheet1
96         SourceFilePath = .Range("E8").Value 'Source Workbook File Path
97         On Error GoTo MissingWorkbook
98         Set SourceWkBk = Workbooks(Dir(SourceFilePath)) 'Set Source Workbook
99         SourceShtName = .Range("I9").Value 'Source Worksheet Name
100    End With
101    With SourceWkBk.Sheets(SourceShtName)
102
103        'Get Last Row Of Data
104        'On Error Resume Next
105        LastRow = .Cells.Find(What:="*", After:=.Range("A1"), _
106            LookAt:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByRows, SearchDirection:=xlPrevious,
107            MatchCase:=False).Row
108        'Get Last Column Of Data
109        LastCol = .Cells.Find(What:="*", After:=.Range("A1"), _
110            LookAt:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByColumns, SearchDirection:=
111            xlPrevious, MatchCase:=False).Column
112        On Error GoTo 0
113
114        'Determine First Row Of Table w/ 70% of the cells filled
115        For TableRow = 1 To LastRow
116            If (Application.WorksheetFunction.CountA(Range(.Cells(TableRow, 1), .Cells(TableRow,
117                LastCol)))) / LastCol) >= 0.7 Then
118                FirstRow = TableRow
119                Exit For
120            End If
121        Next TableRow

```

```
1 2
119
120     'Determine First Column Of Table w/ 50% of the cells filled
121     For TableCol = 1 To LastCol
122         If (Application.WorksheetFunction.CountA(Range(.Cells(1, TableCol), .Cells(LastRow,
123             TableCol))) / LastRow) >= 0.5 Then
124             FirstCol = TableCol
125             Exit For
126         End If
127     Next TableCol
128 End With
129
130 If Sheet1.Range("A6" ).Value = True Then FirstRow = FirstRow + 1
131
132 Sheet1.Range("F12" ).Value = FirstRow 'Set First Row Of Source Data
133 Sheet1.Range("I12" ).Value = LastRow 'Set Last Row Of Source Data
134 Sheet1.Range("F13" ).Value = FirstCol 'Set First Column Of Source Data
135 Sheet1.Range("I13" ).Value = LastCol 'Set Last Column Of Source Source Data
136
137
138 ImportWkBk.Activate
139 Exit Sub
140 MissingWorkbook:
141 MsgBox "Please check for a correct Source Workbook"
142 End Sub
```

—, 34

A

Activate, 35
Add, 33
After, 34
Application, 33-35

B

BrowseDynamicSourceFolder, 33
BrowseForSource, 33
BrowseMoveFilesToFolder, 33

C

Cells, 34, 35
ClearContents, 33
Column, 34
CountA, 34, 35

D

Delete, 33
Dir, 33, 34
DynamicSource, 34
DynSourceFold, 33

E

Explicit, 33

F

FileDialog, 33
FileFldr, 33
Filters, 33
Find, 34
FirstCol, 34, 35
FirstRow, 34, 35

G

GetSourceData, 34

I

ImportWkBk, 34, 35

L

LastCol, 34, 35
LastRow, 34, 35
Lookat, 34
LookIn, 34

M

MatchCase, 34
MissingWorkbook, 34, 35
MsgBox, 35
msoCTrue, 34
msoFalse, 34
msoFileDialogFilePicker, 33
msoFileDialogFolderPicker, 33

N

Name, 33
NoSelection, 33

O

Open, 33

R

Range, 33-35
Row, 34

S

SearchDirection, 34
SearchOrder, 34
SelectedItems, 33
Shapes, 34
Sheet1, 33-35
Sheets, 34
Show, 33
SourceFilePath, 33, 34
SourceShtName, 34
SourceWkBk, 33, 34
SourceWs, 33

T

TableCol, 34, 35
TableRow, 34
ThisWorkbook, 34
Title, 33

V

Validation, 33
Value, 33-35
vbDirectory, 33
Visible, 34

W

What, 34
Workbook, 33, 34
Workbooks, 33, 34
Worksheet, 33
WorksheetFunction, 34, 35
Worksheets, 33
WsRow, 33, 34

X

xlByColumns, 34
xlByRows, 34
xlFormulas, 34
xlPart, 34
xlPrevious, 34

```
1 Option Explicit
2
3 Sub ImportStep1()
4 StopCalc 'Import Step 1
5 ResetHideSteps
6 Sheet1.Shapes("Step1").Fill.ForeColor.RGB = RGB(51, 204, 255)
7 Sheet1.Shapes("Step1").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
8 Sheet1.Range("5:22").EntireRow.Hidden = False
9 Sheet1.Shapes("Step1Grp").Visible = msoCTrue
10 ResetCalc
11 End Sub
12
13 Sub ImportStep2()
14 StopCalc 'Import Step 2
15 ResetHideSteps
16 Sheet1.Range("23:40").EntireRow.Hidden = False
17 Sheet1.Shapes("Step2").Fill.ForeColor.RGB = RGB(51, 204, 255)
18 Sheet1.Shapes("Step2").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
19 Sheet1.Shapes("Step2Grp").Visible = msoCTrue
20 ResetCalc
21 End Sub
22
23 Sub ImportStep3()
24 StopCalc 'Import Step 3
25 ResetHideSteps
26 Sheet1.Range("59:95").EntireRow.Hidden = False
27 Sheet1.Shapes("Step3").Fill.ForeColor.RGB = RGB(51, 204, 255)
28 Sheet1.Shapes("Step3").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
29 Sheet1.Shapes("Step3Grp").Visible = msoCTrue
30 ResetCalc
31 End Sub
32
33 Sub ImportStep4()
34 StopCalc 'Import Step 4
35 ResetHideSteps
36 Sheet1.Range("41:58").EntireRow.Hidden = False
37 Sheet1.Shapes("Step4").Fill.ForeColor.RGB = RGB(51, 204, 255)
38 Sheet1.Shapes("Step4").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(0, 0, 0)
39 Sheet1.Shapes("Step4Grp").Visible = msoCTrue
40 ResetCalc
41 End Sub
42
43 Sub ResetHideSteps()
44 With Sheet1 'Hide All Shapes & Rows
45 .Range("5:165").EntireRow.Hidden = True
46 .Shapes.Range(Array("ButtonSet")).Fill.ForeColor.RGB = RGB(21, 51, 87)
47 .Shapes.Range(Array("ButtonSet")).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(
48 255, 255, 255)
49 .Shapes("Step1Grp").Visible = msoFalse
50 .Shapes("Step2Grp").Visible = msoFalse
51 .Shapes("Step3Grp").Visible = msoFalse
52 .Shapes("Step4Grp").Visible = msoFalse
53 End With
End Sub
```


A

Array, [37](#)

E

EntireRow, [37](#)

Explicit, [37](#)

F

Fill, [37](#)

Font, [37](#)

ForeColor, [37](#)

H

Hidden, [37](#)

I

ImportStep1, [37](#)

ImportStep2, [37](#)

ImportStep3, [37](#)

ImportStep4, [37](#)

M

msoCTrue, [37](#)

msoFalse, [37](#)

R

Range, [37](#)

ResetCalc, [37](#)

ResetHideSteps, [37](#)

RGB, [37](#)

S

Shapes, [37](#)

Sheet1, [37](#)

StopCalc, [37](#)

T

TextFrame2, [37](#)

TextRange, [37](#)

V

Visible, [37](#)

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