VBA SOURCE CODE BOOK

Mastering Drag & Drop Shapes In Excel

DOWNLOAD APPLICATION

VIEW TRAINING

by: Randy Austin

ABOUT THE AUTHOR

A 4-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 over 471,000 YouTube subscribers, 35,448,742 video views, 430+ comprehensive training videos, and a thriving 65,000 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

DISCORD

INSTAGRAM

TELEGRAM

RUMBLE

Microsoft® MVP Most Valuable Professional

OUR COURSES & PRODUCTS

MENTORSHIP EXPRESS

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

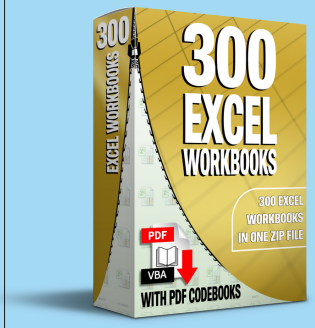
Click here to learn more

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

This incredible 13-hour freelancer masterclass will teach anyone how to be a successful freelancer with my proven 9-Phase 'Financial Freedom Roadmap' and includes 30+ downloads and exercises.

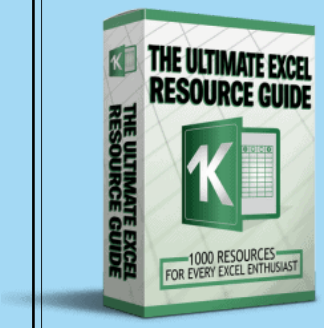
Click here to learn more



Incredible Package of 300 of my BEST Applications with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "300 Workbook Library".

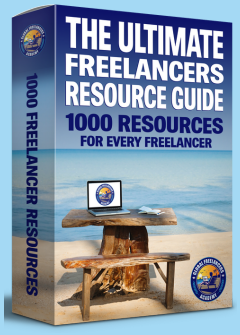
Click here to learn more

OUR COURSES & PRODUCTS



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



Freelancing essentials from freelance tools to freelance templates, in this Ultimate Freelancer's Resource Guide for freelancers at any stage in their career. With 1,000 Live Links, and a single click-to-update application, you will always have the most current and up-to-date information at your fingertips.

Click here to learn more

Revolutionize the way you work with Excel and take productivity to the next level with the Excel AI Toolpack - the FIRST AI tool designed for ANY Windows Desktop version of Excel. This incredible add-in combines FIVE powerful AI tools, transforming your Excel into an intelligent powerhouse!

Click here to learn more

Table of Contents

```
1   Option Explicit
2
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
3   Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
      it in
5     Dim Cal As Shape
6     On Error Resume Next
7     Set Cal = Shapes("Calendar" )
8     On Error GoTo 0
9     If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
10    If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
11
12    If Target.Count > 1 Then Exit Sub
13    If Not Intersect(Target, Range("I3" )) Is Nothing Then 'Change this to any cell (or
      cells) you would like to have the Pop-Up Calendar Appear
14      CheckForSheet
15      CalendarShow
16    Else:
17      CheckForSheet
18      CalendarHide
19    End If
20  End Sub
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
```

```
1    Option Explicit
2
```

```
1   Option Explicit
2
3   Sub Job_CheckForMove()
4     Dim DestRow As Long, DestCol As Long, JobRow As Long, JobCol As Long, CountDelay As
      Long
5     Dim JobID As String, DestEmpl As String
6     Dim DestDate As Date
7     With Schedule
8       JobID = .Range("B2" ).Value 'Select Job ID
9       If .Range("B3" ).Value <> ""  Then JobRow = .Range("B3" ).Value 'Select Job DB
        Row
10      For CountDelay = 1 To 100000
11        DoEvents
12        If .Range("B12" ).Value = True Then GoTo EndDragDrop 'Exit loop move job =
          True
13        With .Shapes("SchedJob"  & JobID)
14          If .Left <> Schedule.Range("B10" ).Value Or .Top <> Schedule.Range("B11" ).
            Value Then 'Shape has been moved
15            'Check to see if shape is moved within proper grid area
16            If .Left < Schedule.Range("E1" ).Left Or .Left > Schedule.Range("M1" ).
              Left Or .Top < Schedule.Range("E6" ).Top - 1 Or Schedule.Range("D"  & .
              TopLeftCell.Row).Value = Empty Then
17              MsgBox "Please move the job shape within the given schedule"
18              GoTo EndDragDrop
19            End If
20
21            If JobID = ""  Then  'On New Job
22              JobRow = JobsDB.Range("A9999" ).End(xlUp).Row + 1 'First Avail Row
23              JobID = Schedule.Range("B4" ).Value
24              Schedule.Range("B2" ).Value = Schedule.Range("B4" ).Value 'Set Next
                ID
25              JobsDB.Range("A"  & JobRow).Value = JobID 'Set Job ID
26              JobsDB.Range("B"  & JobRow).Value = .TextFrame2.TextRange.Text 'Set
                Job Name
27            End If
28
29            If .Left > Schedule.Range("L1" ).Left And .Left < Schedule.Range("M1" ).
              Left Then 'Unscheduled Job
30              DestDate = 0 'Clear Date
31              DestEmpl = ""  'Clear Employee
32            Else 'Scheduled within grid
33              DestEmpl = Schedule.Range("D"  & .TopLeftCell.Row).Value
                'Destination Employee
34              DestDate = Schedule.Cells(5, .TopLeftCell.Column).Value 'Destination
                Date
35            End If
36            JobsDB.Range("C"  & JobRow).Value = DestEmpl 'Update Employee
37            JobsDB.Range("D"  & JobRow).Value = DestDate 'Update Date
38            GoTo EndDragDrop
39          End If
40        End With
41      Next CountDelay
42      EndDragDrop:
43      Schedule_Refresh 'Refresh Schedule
44      Job_Load 'Refresh job data
45      .Range("B12" ).Value = True
46    End With
47  End Sub
```

```
Option Explicit

Dim JobRow As Long, JobCol As Long

Sub Job_Delete()
  If MsgBox("Are you sure you want to delete this Job?" , vbYesNo, "Delete Job" ) =
  vbNo Then Exit Sub
  With Schedule
    If .Range("B3" ).Value = Empty Then GoTo NotSaved
    JobRow = .Range("B3" ).Value 'Job DB Row
    JobsDB.Range(JobRow & ":"  & JobRow).EntireRow.Delete
    NotSaved:
    Job_New
    Schedule_Refresh 'Run macro to refresh Schedule
  End With
End Sub

Sub Job_Load()
  With Schedule
    If .Range("B3" ).Value = Empty Then
      MsgBox "Please select a correct Job to load"
      Exit Sub
    End If
    Application.ScreenUpdating = False
    JobRow = .Range("B3" ).Value 'Job DB Row
    For JobCol = 2 To 5
      .Range(JobsDB.Cells(1, JobCol).Value).Value = JobsDB.Cells(JobRow, JobCol).
      Value 'Bring Over Data
    Next JobCol
    Application.ScreenUpdating = True
  End With
End Sub

Sub Job_New()
  Schedule.Range("B2,E3,G3,I3,K3" ).ClearContents 'Clear Out Existing Cells
End Sub

Sub Job_NewThisWeek()
  Dim JobName As String
  JobName = InputBox("Please enter the job name" , "Enter Job Name" , , 10000, 5000)
  If JobName = ""  Then Exit Sub 'Exit on empty job name
  Schedule.Shapes("SampleJobShp" ).Duplicate.Name = "SchedJob"
  With Schedule.Shapes("SchedJob" )
    .Left = Schedule.Range("G4" ).Left + 10
    .Top = Schedule.Range("G4" ).Top + 2
    .TextFrame2.TextRange.Text = JobName
    .Select
    Schedule.Range("B10" ).Value = .Left
    Schedule.Range("B11" ).Value = .Top
    Schedule.Range("B12" ).Value = False
    Schedule.Range("B2" ).ClearContents 'Clear Job ID
  End With
  MsgBox "Please drag this new job to an employee and date on the schedule"
  Job_CheckForMove
End Sub


Sub Job_SaveUpdate()
  With Schedule
    If Application.WorksheetFunction.CountA(.Range("E3,G3,I3" )) < 3 Then
      MsgBox "Please fill in the required fields, Job Name, Employee & Date"
1 2 3
```

```
60          Exit Sub
61        End If
62        Application.ScreenUpdating = False
63        If .Range("B3" ).Value = Empty Then 'New Job
64          JobRow = JobsDB.Range("A99999" ).End(xlUp).Row + 1  'First Avail. Row
65          .Range("B2" ).Value = .Range("B4" ).Value 'Next Job ID
66          JobsDB.Range("A"  & JobRow).Value = .Range("B2" ).Value 'Job ID
67        Else 'Existing
68          JobRow = .Range("B3" ).Value
69        End If
70
71        For JobCol = 2 To 5
72          JobsDB.Cells(JobRow, JobCol).Value = .Range(JobsDB.Cells(1, JobCol).Value).
            Value 'Bring Over Data
73        Next JobCol
74        Schedule_Refresh 'Run macro to refresh Schedule
75        Application.ScreenUpdating = True
76      End With
77    End Sub
```

```
Dim SelCell As Range
Dim DayName As String



'''''''''''''''''Color Calendar Background'''''''''''''''''''''
Sub CalCol()
  With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
    With Selection.ShapeRange.Fill
      .ForeColor.RGB = ActiveSheet.Shapes(Application.Caller).Fill.ForeColor.RGB
    End With
    ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
  End With
End Sub
Sub CalendarHide()
  Dim DayNum As Long
  'Hide Calendar, Reset Day Colors
  On Error GoTo NoCal
  ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
  Sheets("CalPopUp" ).Range("A7" ).Value = ""
  If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
    For DayNum = 1 To 42
      DayName = DayNum & "Day"
      With ActiveSheet.Shapes(DayName)
        .Fill.ForeColor.RGB = RGB(255, 255, 255)
        .TextFrame2.TextRange.Font.Bold = msoFalse
      End With
    Next DayNum
  End If
  Exit Sub
  NoCal: 'If calendar has been removed by accident, paste in backup calendar from
  CalPopUp Sheet
End Sub

Sub CalendarShow()
  With ActiveSheet
    Set SelCell = ActiveCell
    'Check if active cell is a valid date
    If IsDate(SelCell.Value) = True Then
      Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
    Else: 'If No Date or incorrect Date user current date
      Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
    End If
    'Clear all shapes to white (if calendar is visible)
    If ActiveSheet.Shapes("Calendar" ).Visible = True Then
      For DayNum = 1 To 42
        DayName = DayNum & "Day"
        With ActiveSheet.Shapes(DayName)
          .Fill.ForeColor.RGB = RGB(255, 255, 255)
          .TextFrame2.TextRange.Font.Bold = msoFalse
        End With
      Next DayNum
    End If

    Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).
    Value) 'Set Month
    Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
    Value) 'Set Year
    DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
    ' UnGroupCal
    If InStr(.Shapes("1Day" ).DrawingObject.Formula, "]" ) <> 0 Then   'Run Workbook
```

```
1 2 3
    Link Remover and Cell Link Replacement
59    MacroLinkRemover
60    CalFormulaReplacement
61  End If
62  'GroupCal
63  On Error GoTo NoCal
64  If DayName = "Day"  Then DayName = Day(Date) & "Day"  'Set Default Day
65  With ActiveSheet.Shapes(DayName)
66    .Fill.ForeColor.RGB = RGB(252, 213, 180)
67    .TextFrame2.TextRange.Font.Bold = msoTrue
68  End With
69  On Error GoTo NoCal
70  .Shapes("Calendar" ).Visible = msoCTrue
71  .Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" , "CalCol4" ,
    "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible = False '
72  .Shapes("Calendar" ).Left = SelCell.Left
73  .Shapes("Calendar" ).Placement = xlMove
74  .Shapes("Calendar" ).ZOrder msoBringToFront
75  .Shapes("Calendar" ).Top = SelCell.Offset(1, 0).Top
76  If Sheets("CalPopUp" ).Range("A6" ).Value > 0 Then
77    .Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" , "41Day"
    , "42Day" )).Visible = True
78  Else:
79    .Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" , "41Day"
    , "42Day" )).Visible = False
80  End If
81  Sheets("CalPopUp" ).Range("A7" ).Value = SelCell.Address
82  ActiveCell.Select
83 End With
84 Exit Sub
85 NoCal:
86 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
   calendar over from another sheet and paste into this sheet"
87 End Sub
88 Sub CalFormulaReplacement()
89  With ActiveSheet
90    Dim DayNum, ColNum, RowNum As Long
91    Dim Shp As Shape
92    ColNum = 2
93    RowNum = 1
94    For DayNum = 1 To 42
95      .Shapes(DayNum & "Day" ).DrawingObject.Formula = "=CalPopUp!"  & .Cells(RowNum
      , ColNum).Address
96      ColNum = ColNum + 1
97      If ColNum = 9 Then
98        ColNum = 2
99        RowNum = RowNum + 1
100     End If
101   Next DayNum
102   .Shapes("Month" ).DrawingObject.Formula = "=CalPopUp!$A$4"
103   .Shapes("Year" ).DrawingObject.Formula = "=CalPopUp!$A$2"
104 End With
105 End Sub
106
107 Sub CheckForSheet()
108   'Checks for existance of Calendar Pop-up Worksheet
109   Dim ws As Worksheet
110   On Error GoTo CreateWS
111   Set ws = ActiveWorkbook.Sheets("CalPopUp" )
112   Exit Sub
113   CreateWS:
1
```

```
    1
114     CreateCalSht
115 End Sub
116
117
118 'Create Calendar Sheet on First Run of Calendar
119 Sub CreateCalSht()
120     Dim ColCnt, RowCnt, DayCnt, CalCol As Long
121     Dim ws, ActSht As Worksheet
122     Set ActSht = ActiveSheet
123     'On Error GoTo NoCal
124     ActiveSheet.Shapes("Calendar" ).Copy
125     Set ws = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.
        Count))
126     ws.Name = "CalPopUp"
127     ActSht.Activate
128
129     'Reassign Shape Links & Macros
130     With ActiveSheet
131       UnGroupCal
132       '.Unprotect
133       .Shapes("PrevYr" ).OnAction = "'"  & ActiveWorkbook.Name & "'!PrevYear"
134       .Shapes("NextYr" ).OnAction = "'"  & ActiveWorkbook.Name & "'!NextYear"
135       .Shapes("NextRec" ).OnAction = "'"  & ActiveWorkbook.Name & "'!NextMonth"
136       .Shapes("NextTri" ).OnAction = "'"  & ActiveWorkbook.Name & "'!NextMonth"
137       .Shapes("PrevRec" ).OnAction = "'"  & ActiveWorkbook.Name & "'!PrevMonth"
138       .Shapes("PrevTri" ).OnAction = "'"  & ActiveWorkbook.Name & "'!PrevMonth"
139       .Shapes("SetBtn" ).OnAction = "'"  & ActiveWorkbook.Name & "'!ShowSettings"
140       .Shapes("Month" ).DrawingObject.Formula = "=CalPopUp!A4"
141       .Shapes("Year" ).DrawingObject.Formula = "=CalPopUp!A2"
142       DayCnt = 1
143       For RowCnt = 1 To 6
144         For ColCnt = 2 To 8
145           .Shapes(DayCnt & "Day" ).DrawingObject.Formula = "=CalPopUp!"  & .Cells(
              RowCnt, ColCnt).Address 'Assigned Linked Cell
146           .Shapes(DayCnt & "Day" ).OnAction = "'"  & ActiveWorkbook.Name &
              "'!SelectDay"  'Assign Macro
147           DayCnt = DayCnt + 1
148         Next ColCnt
149       Next RowCnt
150
151       'Assign Color Macros
152       For CalCol = 1 To 9
153         .Shapes("CalCol"  & CalCol).OnAction = "'"  & ActiveWorkbook.Name &
            "'!CalCol"  'Assign Color Macro
154       Next CalCol
155     End With
156
157     With Sheets("CalPopUp" )
158       .Paste
159       .Visible = xlSheetHidden
160
161       'Add in Formulas and Details
162       .Range("A1" ).Value = Date 'Set Current Date
163       .Range("A2" ).Value = Year(Date) 'Set Current Year
164       .Range("A3" ).Value = Month(Date) 'Set Current Month #
165       .Range("A4" ).Value = "=INDEX(CalMonths,A3,)"
166       .Range("A5" ).Value = "=A4&"  & Chr(34) & " "  & Chr(34) & "&CalYear"
167       .Range("A6" ).Value = "=SUM(B6:H6)"
168       .Range("A8" ).Value = "January"
169       .Range("A8" ).AutoFill Destination:=.Range("A8:A19" ), Type:=xlFillDefault
170       .Range("A20" ).Value =
    1 2
```

```
1 2
        "=IFERROR(INDIRECT(ADDRESS(SUMPRODUCT((B1:H6=A1)*ROW(B1:H6))+6,SUMPRODUCT((B1:H6=A
        1)*COLUMN(B1:H6)),1,1)),"  & Chr(34) & Chr(34) & ")"
171     'Set Defined Names
172     ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
173     ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
174
175     'Add in Calendar Formulas
176
177     .Range("B1" ).Value =
        "=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,Ca
        lMonths,0),1),"  & Chr(34) & Chr(34) & ")"
178     .Range("C1" ).Value = "=IF(B1<>"  & Chr(34) & Chr(34) &
        ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A
        $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
179     .Range("D1" ).Value = "=IF(C1<>"  & Chr(34) & Chr(34) &
        ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A
        $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
180     .Range("E1" ).Value = "=IF(D1<>"  & Chr(34) & Chr(34) &
        ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A
        $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
181     .Range("F1" ).Value = "=IF(E1<>"  & Chr(34) & Chr(34) &
        ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A
        $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
182     .Range("G1" ).Value = "=IF(F1<>"  & Chr(34) & Chr(34) &
        ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A
        $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
183     .Range("H1" ).Value = "=IF(G1<>"  & Chr(34) & Chr(34) &
        ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A
        $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
184     .Range("B2" ).Value = "=H1+1"
185     .Range("C2" ).Value = "=B2+1"
186     .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
187     .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
188     .Range("B5" ).Value = "=IF(OR(H4="  & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
         & Chr(34) & Chr(34) & ",H4+1)"
189     .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3,"  & Chr(34) & Chr(34) &
        ",B5+1),"  & Chr(34) & Chr(34) & ")"
190     .Range("B6" ).Value = "=IFERROR(IF(OR(H5="  & Chr(34) & Chr(34) &
        ",MONTH(H5+1)<>$A$3),"  & Chr(34) & Chr(34) & ",H5+1),"  & Chr(34) & Chr(34) &
        ")"
191     .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3,"  & Chr(34) & Chr(34) &
        ",I5+1),"  & Chr(34) & Chr(34) & ")"
192     .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
193
194     'Set format to Single Day
195     .Range("B1:H6" ).NumberFormat = "d"
196
197     'Add in relative Day #'s
198     .Range("B7" ).Value = "1"
199     .Range("C7" ).Value = "2"
200     .Range("B8" ).Value = "8"
201     .Range("C8" ).Value = "9"
202     .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
203     .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
204     GroupCal
205   End With
206   Exit Sub
207   NoCal:
208   MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
      calendar over from another sheet and paste into this sheet"
209 End Sub
```

```
210  Sub GroupCal()
211      ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
212      Selection.ShapeRange.Name = "NextMonth"
213      ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
214      Selection.ShapeRange.Name = "PrevMonth"
215      ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day"  _
216          , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" ,
         "1Day" , _
217          "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" ,
         "6Day" , _
218          "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
         "18Day"  _
219          , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day"
         , _
220          "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
         "Sa" , _
221          "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
         "CalCol3" , _
222          "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" ,
         "PrevMonth" , _
223          "NextMonth" , "NextYr" , "PrevYr" )).Visible = msoCTrue
224      ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day"  _
225      , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day"
         , _
226      "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
227      "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
         "18Day"  _
228      , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
229      "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
         "Sa" , _
230      "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
         "CalCol3" , _
231      "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth"
         , _
232      "NextMonth" , "NextYr" , "PrevYr" )).Select
233      Selection.ShapeRange.Group.Select
234      Selection.ShapeRange.Name = "Calendar"
235      Selection.Name = "Calendar"
236      Selection.Placement = xlMove
237      ActiveSheet.Shapes("Calendar" ).Placement = 2
238  End Sub
239
240
241
242
243  Sub MacroLinkRemover()
244      'PURPOSE: Remove an external workbook reference from all shapes triggering macros
245      'Source: www.ExcelForFreelancers.com
246      Dim Shp As Shape
247      Dim MacroLink, NewLink As String
248      Dim SplitLink As Variant
249
250      For Each Shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
251
252          'Grab current macro link (if available)
253          On Error GoTo NextShp
254          MacroLink = Shp.OnAction
255
```

1 2

```
    'Determine if shape was linking to a macro
    If MacroLink <> ""  And InStr(MacroLink, "!" ) <> 0 Then
      'Split Macro Link at the exclaimation mark (store in Array)
      SplitLink = Split(MacroLink, "!" )

      'Pull text occurring after exclaimation mark
      NewLink = SplitLink(1)

      'Remove any straggling apostrophes from workbook name
      If Right(NewLink, 1) = "'"  Then
        NewLink = Left(NewLink, Len(NewLink) - 1)
      End If

      'Apply New Link
      Shp.OnAction = NewLink
    End If
    NextShp:
  Next Shp
End Sub
Sub NextMonth()
  'Next Month button
  If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
    DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
    With ActiveSheet.Shapes(DayName)
      .Fill.ForeColor.RGB = RGB(255, 255, 255)
      .TextFrame2.TextRange.Font.Bold = msoFalse
    End With
  End If
  With Sheets("CalPopUp" )
    If .Range("A3" ).Value = 12 Then
      .Range("A3" ).Value = 1
      .Range("A2" ).Value = .Range("A2" ).Value + 1
    Else:
      .Range("A3" ).Value = .Range("A3" ).Value + 1
    End If
    If .Range("A6" ).Value > 0 Then
      ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
      "40Day" , "41Day" , "42Day" )).Visible = True
    Else:
      ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
      "40Day" , "41Day" , "42Day" )).Visible = False
    End If
  End With
End Sub

Sub NextYear()
  ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
  "CalPopUp" ).Range("A2" ).Value + 1
End Sub
Sub PrevMonth()
  'Previous Month Button
  If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
    DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
    With ActiveSheet.Shapes(DayName)
      .Fill.ForeColor.RGB = RGB(255, 255, 255)
      .TextFrame2.TextRange.Font.Bold = msoFalse
    End With
  End If
  With Sheets("CalPopUp" )
    If .Range("A3" ).Value = 1 Then
      .Range("A3" ).Value = 12
```

```
1 2 3
314             .Range("A2" ).Value = .Range("A2" ).Value - 1
315           Else:
316             .Range("A3" ).Value = .Range("A3" ).Value - 1
317           End If
318           If .Range("A6" ).Value > 0 Then
319             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
                "40Day" , "41Day" , "42Day" )).Visible = True
320           Else:
321             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
                "40Day" , "41Day" , "42Day" )).Visible = False
322           End If
323         End With
324       End Sub
325
326       Sub PrevYear()
327         ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
            "CalPopUp" ).Range("A2" ).Value - 1
328       End Sub
329       Sub ReplaceCalendar() 'Shape Deleted
330         ThisWorkbook.Sheets("CalPopUp" ).Shapes("Calendar" ).Copy 'Copy From Developers
            sheet
331         ActiveCell.Select 'Select the active Cell
332         ActiveSheet.Paste 'Paste in Calendar
333       End Sub
334
335       '''''''''''''''''''''''''''''''''''''''
336       '''''Select Day Of The Month
337       '''''''''''''''''''''''''''''''''''''''
338       Sub SelectDay()
339         Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
340         DayNumb = Replace(Application.Caller, "Day" , "" )
341         RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
342         ColNumb = DayNumb Mod 7 + 1
343         If ColNumb = 1 Then ColNumb = 8
344         'On Error Resume Next
345         If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
346         ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
            ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
347         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
348         ActiveCell.Offset(0, 1).Select
349       End Sub
350       Sub ShowSettings()
351         'Show or Hide Calendar Settings Panel
352         If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
353           ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
              "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
              = False
354         Else:
355           ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
              "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
              = True
356         End If
357       End Sub
358
359       Sub UnGroupCal()
360         On Error Resume Next
361         ActiveSheet.Shapes("Calendar" ).Ungroup
362         ActiveSheet.Shapes("NextMonth" ).Ungroup
363         ActiveSheet.Shapes("PrevMonth" ).Ungroup
364         On Error GoTo 0
365       End Sub
```

```
1   Option Explicit
2
3   Dim JobRow As Long, JobCol As Long, LastRow As Long, LastResultRow As Long, ResultRow
    As Long
4   Dim JobID As String, JobName As String, JobEmpl As String
5   Dim JobShp As Shape
6   Dim JobDate As Date
7
8
9   Sub Schedule_Job_Select()
10    With Schedule
11      JobID = Replace(Application.Caller, "SchedJob" , "" ) 'Extract ID
12      .Range("B2" ).Value = JobID 'Set Job ID
13      .Range("B10" ).Value = .Shapes(Application.Caller).Left 'Set Left Pos. of
        Current Selected Shape
14      .Range("B11" ).Value = .Shapes(Application.Caller).Top 'Set Top Pos of Sel.
        Shape
15      .Range("B12" ).Value = False 'Set Move Shape To False
16      .Shapes(Application.Caller).Select 'Select On shape
17      Job_Load
18      Job_CheckForMove
19    End With
20  End Sub
21
22  Sub Schedule_NextWeek()
23    Schedule.Range("E5" ).Value = Schedule.Range("E5" ).Value + 7
24    Schedule_Refresh
25  End Sub
26
27  Sub Schedule_PrevWeek()
28    Schedule.Range("E5" ).Value = Schedule.Range("E5" ).Value - 7
29    Schedule_Refresh
30  End Sub
31
32  Sub Schedule_Refresh()
33    'Clear All existing Job shapes
34    For Each JobShp In Schedule.Shapes
35      If InStr(JobShp.Name, "SchedJob" ) > 0 Then JobShp.Delete 'Delete only shapes
        based on job
36    Next JobShp
37    Schedule.Range("B14" ).ClearContents 'Clear Unscheduled Jobs
38    With JobsDB
39      LastRow = .Range("A99999" ).End(xlUp).Row  'Last Row Of data
40      If LastRow < 4 Then Exit Sub
41      .Range("A3:E"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
        "I2:J4" ), CopyToRange:=.Range("L2:P2" ), Unique:=True
42      LastResultRow = .Range("L99999" ).End(xlUp).Row  ' Last Results Row
43      If LastResultRow < 3 Then Exit Sub
44      Application.ScreenUpdating = False
45      For ResultRow = 3 To LastResultRow
46        JobID = .Range("L"  & ResultRow).Value 'Job ID
47        JobName = .Range("M"  & ResultRow).Value 'Job Name
48        JobEmpl = .Range("N"  & ResultRow).Value ' Job Employee
49        JobDate = .Range("O"  & ResultRow).Value 'Job Date
50
51        If JobDate = 0 Or JobName = ""  Then 'Unscheduled / Unassigned
52          JobRow = Schedule.Range("B14" ).Value + 6 'Unscheduled Job Row
53          Schedule.Range("B14" ).Value = Schedule.Range("B14" ).Value + 1
            'Increment Unscheduled Row
54          JobCol = 12 'Set Unscheduled Job Column
55        Else 'Job Is scheduled
    1 2 3 4
```

```
1 2 3 4
56          JobCol = JobDate - [WeekStart] + 5 'Job Column
57          On Error Resume Next
58          JobRow = Schedule.Range("D6:D9999" ).Find(JobEmpl, , xlValues, xlWhole).Row
             'Extract Row
59          On Error GoTo 0
60        End If
61        If JobRow = 0 Then GoTo NextJob
62        Schedule.Shapes("SampleJobShp" ).Duplicate.Name = "SchedJob"  & JobID
          'Duplicate & rename with unique name
63        With Schedule.Shapes("SchedJob"  & JobID)
64          .Left = Schedule.Cells(JobRow, JobCol).Left 'Set Left Pos.
65          .Top = Schedule.Cells(JobRow, JobCol).Top 'Set Top Position
66          .Width = Schedule.Cells(JobRow, JobCol).Width - 2 'Set WIdth Of Shape
67          .Height = Schedule.Cells(JobRow, JobCol).Height - 1 'Set height of Shape
68          .TextFrame2.TextRange.Text = JobName 'Set Job Name as shape text
69          .OnAction = "Schedule_Job_Select"  'Macro To Run
70        End With
71        NextJob:
72      Next ResultRow
73      Application.ScreenUpdating = True
74    End With
75  End Sub
```

Y

Z

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