

# VBA SOURCE CODE BOOK

**EMPLOYEE JOB SCHEDULING**

Job Name: Jack Inspection | Employee: Peter Parker | Job Date: Saturday, January 6, 2024

Employees	Monday, January 1, 2024	Tuesday, January 2, 2024	Wednesday, January 3, 2024	Thursday, January 4, 2024	Friday, January 5, 2024	Saturday, January 6, 2024	Sunday, January 7, 2024	UNSCHEDULED / UNASSIGNED
Fred Fredders		Jim's Kitchen Remodel						Lisa's Kitchen
Dave Davidson	Frank's Garage					in Kitchen		
Peter Parker						Inspection		
Mary Smith								
Mark Mason								
Jack Johnson								
Lisa Mathews			Fredders House Fix					
Greg Perkins								

**FEATURE, FIX OR FOCUS**

**DRAG & DROP SHAPES**

**UPDATED**

**EXCLUSIVE FOR SILVER & GOLD MEMBERS**

**Excel For Freelancers**

## 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®**  
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](#)

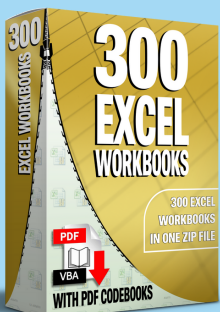
-



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](#)

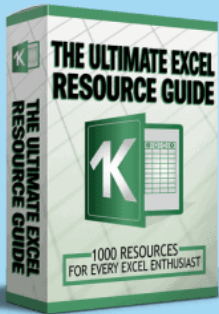
-



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](#)

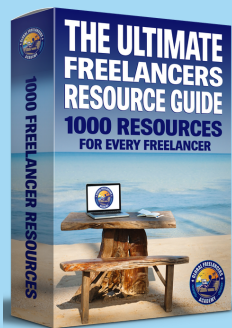
# OUR COURSES & PRODUCTS



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

Projects.....	2
VBAProject.....	2
Documents.....	2
EmpDB.....	2
(Declarations).....	2
JobsDB.....	3
(Declarations).....	3
ProcessSteps.....	4
(Declarations).....	4
Schedule.....	5
(Declarations).....	5
Worksheet_SelectionChange [Sub ].....	5
Sheet1.....	6
(Declarations).....	6
Sheet3.....	7
(Declarations).....	7
ThisWorkbook.....	8
(Declarations).....	8
Modules.....	9
DragDrop_Macros.....	9
(Declarations).....	9
Job_CheckForMove [Sub ].....	9
Job_Macros.....	10
(Declarations).....	10
Job_Delete [Sub ].....	10
Job_Load [Sub ].....	10
Job_New [Sub ].....	10
Job_NewThisWeek [Sub ].....	10
Job_SaveUpdate [Sub ].....	10
PopUp_Calendar.....	12
(Declarations).....	12
CalCol [Sub ].....	12
CalendarHide [Sub ].....	12
CalendarShow [Sub ].....	12
CalFormulaReplacement [Sub ].....	13
CheckForSheet [Sub ].....	13
CreateCalSht [Sub ].....	14
GroupCal [Sub ].....	16
MacroLinkRemover [Sub ].....	16
NextMonth [Sub ].....	17
NextYear [Sub ].....	17
PrevMonth [Sub ].....	17
PrevYear [Sub ].....	18
ReplaceCalendar [Sub ].....	18
SelectDay [Sub ].....	18
ShowSettings [Sub ].....	18
UnGroupCal [Sub ].....	18
Schedule_Macros.....	19
(Declarations).....	19
Schedule_Job_Select [Sub ].....	19
Schedule_NextWeek [Sub ].....	19
Schedule_PrevWeek [Sub ].....	19
Schedule_Refresh [Sub ].....	19

1  
2

Option Explicit

1 Option Explicit

2

1  
2

Option Explicit



```
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  
2

Option Explicit

1  
2

Option Explicit

1  
2

Option Explicit

```

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

```

```

1 Option Explicit
2
3 Dim JobRow As Long, JobCol As Long
4
5 Sub Job_Delete()
6     If MsgBox("Are you sure you want to delete this Job?" , vbYesNo, "Delete Job" ) =
7         vbNo Then Exit Sub
8     With Schedule
9         If .Range("B3" ).Value = Empty Then GoTo NotSaved
10        JobRow = .Range("B3" ).Value 'Job DB Row
11        JobsDB.Range(JobRow & ":" & JobRow).EntireRow.Delete
12        NotSaved:
13        Job_New
14        Schedule_Refresh 'Run macro to refresh Schedule
15    End With
16 End Sub
17
18 Sub Job_Load()
19     With Schedule
20         If .Range("B3" ).Value = Empty Then
21             MsgBox "Please select a correct Job to load"
22             Exit Sub
23         End If
24         Application.ScreenUpdating = False
25         JobRow = .Range("B3" ).Value 'Job DB Row
26         For JobCol = 2 To 5
27             .Range(JobsDB.Cells(1, JobCol).Value).Value = JobsDB.Cells(JobRow, JobCol).
28                 Value 'Bring Over Data
29         Next JobCol
30         Application.ScreenUpdating = True
31     End With
32 End Sub
33
34 Sub Job_New()
35     Schedule.Range("B2,E3,G3,I3,K3" ).ClearContents 'Clear Out Existing Cells
36 End Sub
37
38 Sub Job_NewThisWeek()
39     Dim JobName As String
40     JobName = InputBox("Please enter the job name" , "Enter Job Name" , , 10000, 5000)
41     If JobName = "" Then Exit Sub 'Exit on empty job name
42     Schedule.Shapes("SampleJobShp" ).Duplicate.Name = "SchedJob"
43     With Schedule.Shapes("SchedJob" )
44         .Left = Schedule.Range("G4" ).Left + 10
45         .Top = Schedule.Range("G4" ).Top + 2
46         .TextFrame2.TextRange.Text = JobName
47         .Select
48         Schedule.Range("B10" ).Value = .Left
49         Schedule.Range("B11" ).Value = .Top
50         Schedule.Range("B12" ).Value = False
51         Schedule.Range("B2" ).ClearContents 'Clear Job ID
52     End With
53     MsgBox "Please drag this new job to an employee and date on the schedule"
54     Job_CheckForMove
55 End Sub
56
57 Sub Job_SaveUpdate()
58     With Schedule
59         If Application.WorksheetFunction.CountA(.Range("E3,G3,I3" )) < 3 Then
60             MsgBox "Please fill in the required fields, Job Name, Employee & Date"
61         End If
62     End With
63 End Sub

```

1 2 3

```
60      1 2 3      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).
73          Value 'Bring Over Data
74      Next JobCol
75      Schedule_Refresh 'Run macro to refresh Schedule
76      Application.ScreenUpdating = True
77      End With
78      End Sub
```

```

1 Dim SelCell As Range
2 Dim DayName As String
3
4
5
6 .....''Color Calendar Background''.....
7 Sub CalCol()
8     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
9         With Selection.ShapeRange.Fill
10            .ForeColor.RGB = ActiveSheet.Shapes(Application.Caller).Fill.ForeColor.RGB
11        End With
12        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
13    End With
14 End Sub
15 Sub CalendarHide()
16     Dim DayNum As Long
17     'Hide Calendar, Reset Day Colors
18     On Error GoTo NoCal
19     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
20     Sheets("CalPopUp" ).Range("A7" ).Value = ""
21     If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
22         For DayNum = 1 To 42
23             DayName = DayNum & "Day"
24             With ActiveSheet.Shapes(DayName)
25                 .Fill.ForeColor.RGB = RGB(255, 255, 255)
26                 .TextFrame2.TextRange.Font.Bold = msoFalse
27             End With
28         Next DayNum
29     End If
30     Exit Sub
31     NoCal: 'If calendar has been removed by accident, paste in backup calendar from
32         CalPopUp Sheet
33 End Sub
34 Sub CalendarShow()
35     With ActiveSheet
36         Set SelCell = ActiveCell
37         'Check if active cell is a valid date
38         If IsDate(SelCell.Value) = True Then
39             Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
40         Else: 'If No Date or incorrect Date user current date
41             Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
42         End If
43         'Clear all shapes to white (if calendar is visible)
44         If ActiveSheet.Shapes("Calendar" ).Visible = True Then
45             For DayNum = 1 To 42
46                 DayName = DayNum & "Day"
47                 With ActiveSheet.Shapes(DayName)
48                     .Fill.ForeColor.RGB = RGB(255, 255, 255)
49                     .TextFrame2.TextRange.Font.Bold = msoFalse
50                 End With
51             Next DayNum
52         End If
53
54         Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).
55         Value) 'Set Month
56         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
57         Value) 'Set Year
58         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
59         ' UnGroupCal
60         If InStr(.Shapes("1Day" ).DrawingObject.Formula, "]" ) <> 0 Then 'Run Workbook

```

1 2 3



```

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",
72 "CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible = False '
73 .Shapes("Calendar").Left = SelCell.Left
74 .Shapes("Calendar").Placement = xlMove
75 .Shapes("Calendar").ZOrder msoBringToFront
76 .Shapes("Calendar").Top = SelCell.Offset(1, 0).Top
77 If Sheets("CalPopUp").Range("A6").Value > 0 Then
78     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
79 "42Day")).Visible = True
80 Else:
81     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
82 "42Day")).Visible = False
83 End If
84 Sheets("CalPopUp").Range("A7").Value = SelCell.Address
85 ActiveCell.Select
86 End With
87 Exit Sub
88 NoCal:
89 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
90 calendar over from another sheet and paste into this sheet"
91 End Sub
92
93 Sub CalFormulaReplacement()
94 With ActiveSheet
95     Dim DayNum, ColNum, RowNum As Long
96     Dim Shp As Shape
97     ColNum = 2
98     RowNum = 1
99     For DayNum = 1 To 42
100         .Shapes(DayNum & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(RowNum
101         , ColNum).Address
102         ColNum = ColNum + 1
103         If ColNum = 9 Then
104             ColNum = 2
105             RowNum = RowNum + 1
106         End If
107     Next DayNum
108     .Shapes("Month").DrawingObject.Formula = "=CalPopUp!$A$4"
109     .Shapes("Year").DrawingObject.Formula = "=CalPopUp!$A$2"
110 End With
111 End Sub
112
113 Sub CheckForSheet()
114 'Checks for existance of Calendar Pop-up Worksheet
115 Dim ws As Worksheet
116 On Error GoTo CreateWS
117 Set ws = ActiveWorkbook.Sheets("CalPopUp")
118 Exit Sub
119 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
171 1)*COLUMN(B1:H6)),1,1)), " & Chr(34) & Chr(34) & ")
'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$4,Ca
178 lMonths,0),1)," & Chr(34) & Chr(34) & ")
.Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A
179 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")")
.Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A
180 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")")
.Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A
181 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")")
.Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A
182 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")")
.Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A
183 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")")
.Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A
184 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")")
.Range("B2" ).Value = "=H1+1"
185 .Range("C2" ).Value = "=B2+1"
186 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
187 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
188 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$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" ,
217         "1Day" , _
218         "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" ,
219         "6Day" , _
220         "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
221         "18Day" _
222         , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day"
223         , _
224         "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
225         "Sa" , _
226         "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
227         "CalCol3" , _
228         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" ,
229         "PrevMonth" , _
230         "NextMonth" , "NextYr" , "PrevYr" )).Visible = msoCTrue
231     ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
232         , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day"
233         , _
234         "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
235         "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
236         "18Day" _
237         , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
238         "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
239         "Sa" , _
240         "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
241         "CalCol3" , _
242         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth"
243         , _
244         "NextMonth" , "NextYr" , "PrevYr" )).Select
245     Selection.ShapeRange.Group.Select
246     Selection.ShapeRange.Name = "Calendar"
247     Selection.Name = "Calendar"
248     Selection.Placement = xlMove
249     ActiveSheet.Shapes("Calendar" ).Placement = 2
250 End Sub

```

```

251 Sub MacroLinkRemover()
252     'PURPOSE: Remove an external workbook reference from all shapes triggering macros
253     'Source: www.ExcelForFreelancers.com
254     Dim Shp As Shape
255     Dim MacroLink, NewLink As String
256     Dim SplitLink As Variant
257
258     For Each Shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
259
260         'Grab current macro link (if available)
261         On Error GoTo NextShp
262         MacroLink = Shp.OnAction

```

```

1 2
256 'Determine if shape was linking to a macro
257 If MacroLink <> "" And InStr(MacroLink, "!") <> 0 Then
258 'Split Macro Link at the exclamation mark (store in Array)
259 SplitLink = Split(MacroLink, "!")
260
261 'Pull text occurring after exclamation mark
262 NewLink = SplitLink(1)
263
264 'Remove any stragglng apostrophes from workbook name
265 If Right(NewLink, 1) = "'" Then
266     NewLink = Left(NewLink, Len(NewLink) - 1)
267 End If
268
269 'Apply New Link
270 Shp.OnAction = NewLink
271 End If
272 NextShp:
273 Next Shp
274 End Sub
275 Sub NextMonth()
276 'Next Month button
277 If Sheets("CalPopUp").Range("A20").Value <> Empty Then
278     DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
279     With ActiveSheet.Shapes(DayName)
280         .Fill.ForeColor.RGB = RGB(255, 255, 255)
281         .TextFrame2.TextRange.Font.Bold = msoFalse
282     End With
283 End If
284 With Sheets("CalPopUp")
285     If .Range("A3").Value = 12 Then
286         .Range("A3").Value = 1
287         .Range("A2").Value = .Range("A2").Value + 1
288     Else:
289         .Range("A3").Value = .Range("A3").Value + 1
290     End If
291     If .Range("A6").Value > 0 Then
292         ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day",
293             "40Day", "41Day", "42Day")).Visible = True
294     Else:
295         ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day",
296             "40Day", "41Day", "42Day")).Visible = False
297     End If
298 End With
299 End Sub
300 Sub NextYear()
301     ThisWorkbook.Sheets("CalPopUp").Range("A2").Value = ThisWorkbook.Sheets(
302         "CalPopUp").Range("A2").Value + 1
303 End Sub
304 Sub PrevMonth()
305 'Previous Month Button
306 If Sheets("CalPopUp").Range("A20").Value <> Empty Then
307     DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
308     With ActiveSheet.Shapes(DayName)
309         .Fill.ForeColor.RGB = RGB(255, 255, 255)
310         .TextFrame2.TextRange.Font.Bold = msoFalse
311     End With
312 End If
313 With Sheets("CalPopUp")
314     If .Range("A3").Value = 1 Then
315         .Range("A3").Value = 12

```

1 2 3

```

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



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



—  
\_, 16

**A**

Activate, 14  
ActiveCell, 12, 13, 18  
ActiveSheet, 12-14, 16-18  
ActiveWorkbook, 13-15  
ActSht, 14  
Add, 14, 15  
Address, 13, 14  
AdvancedFilter, 19  
After, 14  
Application, 10-12, 18-20  
Array, 12, 13, 16-18  
AutoFill, 14, 15

**B**

Bold, 12, 13, 17

**C**

Cal, 5  
CalCol, 12, 14  
CalendarHide, 5, 12  
CalendarShow, 5, 12  
CalFormulaReplacement, 13  
Caller, 12, 18, 19  
Cells, 9-11, 13, 14, 18, 20  
CheckForSheet, 5, 13  
Chr, 14, 15  
ClearContents, 10, 19  
ColCnt, 14  
ColNum, 13  
ColNumb, 18  
Column, 9  
Copy, 14, 18  
CopyToRange, 19  
Count, 5, 14  
CountA, 10  
CountDelay, 9  
CreateCalSht, 14  
CreateWS, 13  
CriteriaRange, 19

**D**

Day, 13  
DayCnt, 14  
DayName, 12, 13, 17  
DayNum, 12, 13  
DayNumb, 18  
Delete, 10, 19  
DestCol, 9  
DestDate, 9  
DestEmpl, 9  
Destination, 14, 15  
DestRow, 9  
DoEvents, 9  
DrawingObject, 12-14  
Duplicate, 10, 20

**E**

Empty, 9-12, 17, 18  
EndDragDrop, 9  
EntireRow, 10  
Explicit, 2-10, 19

**F**

Fill, 12, 13, 17  
Find, 20

Font, 12, 13, 17  
ForeColor, 12, 13, 17  
Formula, 12-14

**G**

Group, 16  
GroupCal, 15, 16

**H**

Height, 20

**I**

InputBox, 10  
InStr, 12, 17, 19  
Intersect, 5  
IsDate, 12

**J**

Job\_CheckForMove, 9, 10, 19  
Job\_Delete, 10  
Job\_Load, 9, 10, 19  
Job\_New, 10  
Job\_NewThisWeek, 10  
Job\_SaveUpdate, 10  
JobCol, 9-11, 19, 20  
JobDate, 19, 20  
JobEmpl, 19, 20  
JobID, 9, 19, 20  
JobName, 10, 19, 20  
JobRow, 9-11, 19, 20  
JobsDB, 9-11, 19  
JobShp, 19

**L**

LastResultRow, 19  
LastRow, 19  
Left, 9, 10, 13, 17, 19, 20  
Len, 17

**M**

MacroLink, 16, 17  
MacroLinkRemover, 13, 16  
Month, 12, 14  
MsgBox, 9, 10, 13, 15  
msoBringToFront, 13  
msoCTrue, 13, 16  
msoFalse, 12, 17, 18  
msoTrue, 13

**N**

Name, 10, 14-16, 19, 20  
Names, 15  
NewLink, 16, 17  
NextJob, 20  
NextMonth, 17  
NextShp, 16, 17  
NextYear, 17  
NoCal, 12, 13, 15  
NotSaved, 10  
NumberFormat, 15

**O**

Offset, 13, 18  
OnAction, 14, 16, 17, 20

**P**

Paste, 14, 18  
Placement, 13, 16  
PrevMonth, 17  
PrevYear, 18

**R**

Range, 5, 9-20  
RefersTo, 15  
Replace, 18, 19  
ReplaceCalendar, 5, 18  
ResultRow, 19, 20  
RGB, 12, 13, 17  
Right, 17  
RoundUp, 18  
Row, 9, 11, 19, 20  
RowCnt, 14  
RowNum, 13  
RowNumb, 18

**S**

Schedule, 9, 10, 19, 20  
Schedule\_Job\_Select, 19  
Schedule\_NextWeek, 19  
Schedule\_PrevWeek, 19  
Schedule\_Refresh, 9-11, 19  
ScreenUpdating, 10, 11, 19, 20  
SelCell, 12, 13  
SelectDay, 18  
Selection, 12, 16  
Shape, 5, 13, 16, 19  
ShapeRange, 12, 16  
Shapes, 5, 9, 10, 12-14, 16-20  
Sheets, 12-14, 17, 18  
ShowSettings, 18  
Shp, 13, 16, 17  
Split, 17  
SplitLink, 16, 17

**T**

Target, 5  
Text, 9, 10, 20  
TextFrame2, 9, 10, 12, 13, 17, 20  
TextRange, 9, 10, 12, 13, 17, 20  
ThisWorkbook, 14, 17, 18  
Top, 9, 10, 13, 19, 20  
TopLeftCell, 9

**U**

Ungroup, 18  
UnGroupCal, 14, 18  
Unique, 19

**V**

Value, 9-15, 17-19  
vbNo, 10  
vbYesNo, 10  
Visible, 5, 12-14, 16-18

**W**

WeekStart, 20  
Width, 20  
Worksheet, 13, 14  
Worksheet\_SelectionChange, 5  
WorksheetFunction, 10, 18  
ws, 13, 14

**X**

xlFillDefault, 14, 15  
xlFilterCopy, 19  
xlMove, 13, 16  
xlSheetHidden, 14  
xlUp, 9, 11, 19  
xlValues, 20  
xlWhole, 20

**Y**

Year, [12](#), [14](#)

**Z**

ZOrder, [13](#)

# 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*