

# VBA SOURCE CODE BOOK



Learn How To Create &  
Customize Your Own  
Productivity Tracker Today



[DOWNLOAD  
APPLICATION](#)



[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 over 442,000 YouTube subscribers, 33,000,000 video views, 250+ comprehensive training videos, and a thriving 60,000 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**<sup>®</sup>  
Most Valuable  
Professional



# OUR COURSES & PRODUCTS



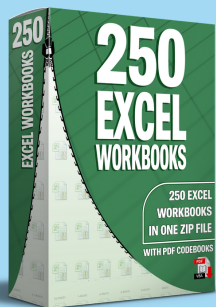
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[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 250 of my BEST Applications now with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "250 Workbook Library".

[Click here to learn more](#)



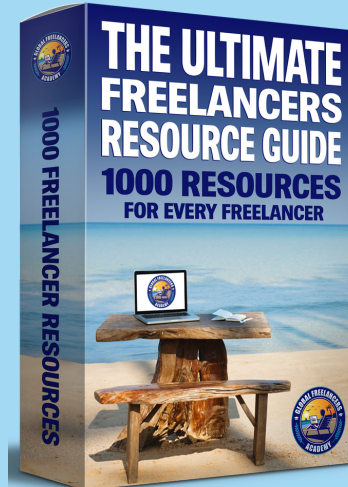
With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)



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 along with a ton of resources for anyone to reach their goals through freelancing.

[Click here to learn more](#)



I've put together all the 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](#)

**Table of Contents**

Projects.....	3
VBAProject.....	3
Documents.....	3
Admin.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub ].....	3
DashData.....	4
(Declarations).....	4
Employees.....	5
(Declarations).....	5
Main.....	6
(Declarations).....	6
Hover1_MouseMove [Sub ].....	6
Hover2_MouseMove [Sub ].....	6
Hover3_MouseMove [Sub ].....	6
Hover4_MouseMove [Sub ].....	6
Worksheet_Change [Sub ].....	6
Worksheet_SelectionChange [Sub ].....	6
Sheet1.....	8
(Declarations).....	8
Tasks.....	9
(Declarations).....	9
ThisWorkbook.....	10
(Declarations).....	10
Modules.....	11
AdminMacs.....	11
(Declarations).....	11
Admin_SetStatusColor [Sub ].....	11
BrowseForEmpIPicFolder [Sub ].....	11
Dashboard_Macros.....	12
(Declarations).....	12
Dashboard_Refresh [Sub ].....	12
DateRange_Change [Sub ].....	12
MainScreen_Macs.....	13
(Declarations).....	13
Employee_Sort [Sub ].....	13
Menu_Change [Sub ].....	13
Menu_ColorDefault [Sub ].....	13
Tasks_AddAttach [Sub ].....	14
PopUp_Calendar.....	15
(Declarations).....	15
CalCol [Sub ].....	15
CalendarHide [Sub ].....	15
CalendarShow [Sub ].....	15
CalFormulaReplacement [Sub ].....	16
CheckForSheet [Sub ].....	16
CreateCalSht [Sub ].....	17
GroupCal [Sub ].....	19
MacroLinkRemover [Sub ].....	19
NextMonth [Sub ].....	20
NextYear [Sub ].....	20
PrevMonth [Sub ].....	20
PrevYear [Sub ].....	21
ReplaceCalendar [Sub ].....	21
SelectDay [Sub ].....	21
ShowSettings [Sub ].....	21
UnGroupCal [Sub ].....	21
Schedule_Macros.....	23
(Declarations).....	23
Sched_CheckForMove [Sub ].....	23
Sched_GoToTask [Sub ].....	23
Sched_NextDay [Sub ].....	24
Sched_PrevDay [Sub ].....	24
Sched_SelectTask [Sub ].....	24
Sched_Today [Sub ].....	24
Schedule_Refresh [Sub ].....	24

## Table of Contents

---

Task_Macros .....	26
(Declarations) .....	26
Task_Delete [Sub ] .....	26
Task_ListTasks [Sub ] .....	26
Task_Load [Sub ].....	26
Task_MakeRecurring [Sub ] .....	26
Task_New [Sub ] .....	27
Task_SaveUpdate [Sub ].....	27

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5     If Shapes("ColorPalette").Visible = True Then Shapes("ColorPalette").Visible =
        msoFalse
6     'Show color palette
7     If Not Intersect(Target, Range("07:017")) Is Nothing And Range("N" & Target.Row).
        Value <> Empty Then
8         With Shapes("ColorPalette")
9             .Left = Range("0" & Target.Row).Left
10            .Top = Range("0" & Target.Row + 1).Top
11            .Visible = msoCTrue
12        End With
13    End If
14 End Sub
15
```

1  
2

Option Explicit



1  
2

Option Explicit

```
1 Option Explicit
```

```
2
3
4 Private Sub Hover1_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X
  As Single, ByVal Y As Single)
5     Menu_ColorDefault
6     Main.Shapes("Menu1" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 0
  )
7 End Sub
```

```
8
9 Private Sub Hover2_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X
  As Single, ByVal Y As Single)
10    Menu_ColorDefault
11    Main.Shapes("Menu2" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 0
  )
12 End Sub
```

```
13 Private Sub Hover3_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X
  As Single, ByVal Y As Single)
14    Menu_ColorDefault
15    Main.Shapes("Menu3" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 0
  )
16 End Sub
```

```
17 Private Sub Hover4_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X
  As Single, ByVal Y As Single)
18    Menu_ColorDefault
19    Main.Shapes("Menu4" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 0
  )
20 End Sub
```

```
21
22
23
24
25
26 Private Sub Worksheet_Change(ByVal Target As Range)
27     'On Dashboard Date Range Change
28     If Not Intersect(Target, Range("H3" )) Is Nothing And Range("H3" ).Value <> Empty
29         And Range("B8" ).Value = False Then
30             DateRange_Change
31             Dashboard_Refresh
32         End If
```

```
33     'On Completion Metric Change
34     If Not Intersect(Target, Range("Y11" )) Is Nothing Then
35         Range("AB11" ).Validation.Delete
36         On Error Resume Next
37         If Target.Value = "Time" Then Range("AB11" ).Validation.Add xlValidatelist, , ,
38             "=TimeUnits"
39         On Error GoTo 0
40     End If
```

```
41     'On Search Field Change
42     If Not Intersect(Target, Range("U3" )) Is Nothing Then Task_ListTasks
43
44     'On Schedule Date Change
45     If Not Intersect(Target, Range("AI3" )) Is Nothing Then Schedule_Refresh
46
47 End Sub
```

```
48
49 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
50     If Target.CountLarge > 3 Then Exit Sub 'If user selects more than 1 cell, then exit
  sub
```

1

```
1
51 Dim Cal As Shape
52 On Error Resume Next
53 Set Cal = Shapes("Calendar" )
54 On Error GoTo 0
55 If Cal Is Nothing Then ReplaceCalendar 'Shape Delete Copy from Popup Sheet
56 If Cal.Visible = True Then CalendarHide 'Hide Calendar if visible
57 If Not Intersect(Target, Range("J3,L3,Y7,Y9,AB18,AD18,AI3" )) Is Nothing Then
58     CheckForSheet
59     CalendarShow 'Macro to display Calenar
60 End If
61
62 'On Task Selection ensure task ID is present
63 If Not Intersect(Target, Range("R6:V9999" )) Is Nothing And Range("Q" & Target.Row)
64     .Value <> Empty Then
65     Range("B3" ).Value = Target.Row
66     Range("B4" ).Value = Range("Q" & Target.Row).Value 'Task ID
67     Task_Load
68 End If
69 End Sub
```

1 Option Explicit

2

1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2
3 Sub Admin_SetStatusColor()
4     ActiveCell.Interior.Color = Admin.Shapes(Application.Caller).Fill.ForeColor.RGB
5     Admin.Shapes("ColorPalette").Visible = msoFalse
6 End Sub
7
8 Sub BrowseForEmplPicFolder()
9     Dim JobFolder As FileDialog
10    Set JobFolder = Application.FileDialog(msoFileDialogFolderPicker)
11    With JobFolder
12        .Title = "Browse For Employee Pictures Folder"
13        .AllowMultiSelect = False
14        If .Show <> -1 Then GoTo NoSelection
15        Admin.Range("D3").Value = .SelectedItems(1)
16    End With
17    NoSelection:
18 End Sub
```

```
1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, ResultRow As Long
3
4 Sub Dashboard_Refresh()
5     With Tasks
6         DashData.Range("G12:G99").ClearContents 'Clear existing Unit Types, Employees
7         LastRow = .Range("A99999").End(xlUp).Row
8         If LastRow < 4 Then Exit Sub
9         'Get Unique Unit Types
10        .Range("K3:L" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
11        "AV2:AV3"), CopyToRange:=.Range("AX2"), Unique:=True
12        LastResultRow = .Range("AX99999").End(xlUp).Row
13        DashData.Range("G12:G" & LastResultRow + 9).Value = .Range("AX3:AX" &
14        LastResultRow).Value 'Bring Over Unique Units
15    End With
16 End Sub
17 Sub DateRange_Change()
18     Dim RangeRow As Long
19     With Main
20         On Error Resume Next
21         RangeRow = Admin.Range("DateRange").Find(.Range("H3").Value, , xlValues,
22         xlWhole).Row
23         On Error GoTo 0
24         If RangeRow = 0 Then
25             MsgBox "Please select a correct date range from the drop down list."
26             .Range("H3").Select
27             Exit Sub
28         End If
29         .Range("B8").Value = True 'Set Date Range change to true
30         .Range("J3").Value = Admin.Range("G" & RangeRow).Value 'From Date
31         .Range("L3").Value = Admin.Range("H" & RangeRow).Value 'To Date
32         .Range("B8").Value = False 'Set Date Range to False
33     End With
34 End Sub
```



```

1 Option Explicit
2 Dim MenuNum As Long, BtnNum As Long
3
4 Sub Employee_Sort()
5     Dim LastRow As Long
6     With Employees
7         .Range("O3:P9999").ClearContents 'Clear Prior results
8         LastRow = .Range("A99999").End(xlUp).Row
9         If LastRow < 4 Then Exit Sub
10        .Range("B3:B" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
11        "O2"), Unique:=True 'Unique employees
12        With .Sort
13            .SortFields.Clear
14            .SortFields.Add Key:=Tasks.Range("O3"), SortOn:=xlSortOnValues, Order:=
15            xlAscending, DataOption:=xlSortNormal 'Sort
16            .SetRange Tasks.Range("O3:O" & LastRow - 1) 'Set Range
17            .Apply 'Apply Sort
18        End With
19        .Range("P3:P" & LastRow - 1).Value = .Range("O3:O" & LastRow - 1).Value
20        'Create Employee List with [All Employees]
21    End With
22 End Sub
23
24
25 Sub Menu_Change()
26     With Main
27         If Application.Caller = "GoToTaskBtn" Then
28             MenuNum = 2 'Set Default on Go To Task
29         Else
30             MenuNum = Replace(Application.Caller, Left(Application.Caller, 4), "" )
31         End If
32         Menu_ColorDefault
33         Main.Shapes("Menu" & MenuNum).TextFrame2.TextRange.Font.Fill.ForeColor.RGB =
34         RGB(255, 255, 0)
35         .Range("B2").Value = MenuNum
36         .Shapes("MenuMarker").Top = .Shapes("Menu" & MenuNum).Top 'Place menu marker
37         'Hide all Shapes & Columns
38         .Shapes("TaskGrp").Visible = msoFalse 'Hide Task Group
39         .Shapes("SchedGrp").Visible = msoFalse 'Hide Schedule Group
40         .Shapes("DashGrp").Visible = msoFalse 'Hide Dashboard Group
41         .Range("F:CL").EntireColumn.Hidden = True 'Hide all columns
42         Select Case MenuNum
43             Case Is = 1 'Dashboard
44                 .Range("F:P").EntireColumn.Hidden = False 'Show Columns
45                 .Shapes("DashGrp").Visible = msoCTrue 'Show Dashboard Group
46             Case Is = 2 'Tasks
47                 .Range("Q:AF").EntireColumn.Hidden = False 'Show Columns
48                 .Shapes("TaskGrp").Visible = msoCTrue 'Show Task Group
49             Case Is = 3 'Schedule
50                 .Range("AG:BL").EntireColumn.Hidden = False 'Show Columns
51                 .Shapes("SchedGrp").Visible = msoCTrue 'Show Schedule Group
52             Case Is = 4 'Employees
53                 .Range("BZ:CL").EntireColumn.Hidden = False 'Show Columns
54         End Select
55     End With
56 End Sub
57 Sub Menu_ColorDefault()
58     For BtnNum = 1 To 4

```

```
58      1 2 Main.Shapes("Menu" & BtnNumb).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB  
      (255, 255, 255) ↵  
59      Next BtnNumb  
60      End Sub  
61  
62  
63      Sub Tasks_AddAttach()  
64          Dim AttFile As FileDialog  
65          Set AttFile = Application.FileDialog(msoFileDialogFilePicker)  
66          With AttFile  
67              .Title = "Please select a file to attach"  
68              .Filters.Add "Attachment Files", "*.*, 1"  
69              .AllowMultiSelect = False  
70              If .Show <> -1 Then GoTo NoSelection  
71              Main.Range("Y15").Value = .SelectedItems(1)  
72              NoSelection:  
73          End With  
74      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     CalendarHide
36     With ActiveSheet
37         Set SelCell = ActiveCell
38         'Check if active cell is a valid date
39         If IsDate(SelCell.Value) = True Then
40             Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
41         Else: 'If No Date or incorrect Date user current date
42             Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
43         End If
44         'Clear all shapes to white (if calendar is visible)
45         If ActiveSheet.Shapes("Calendar" ).Visible = True Then
46             For DayNum = 1 To 42
47                 DayName = DayNum & "Day"
48                 With ActiveSheet.Shapes(DayName)
49                     .Fill.ForeColor.RGB = RGB(255, 255, 255)
50                     .TextFrame2.TextRange.Font.Bold = msoFalse
51                 End With
52             Next DayNum
53         End If
54
55         Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).
56         Value) 'Set Month
57         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
58         Value) 'Set Year
59         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
60         ' UnGroupCal

```

1 2

```

1 2
59 If InStr(.Shapes("1Day").DrawingObject.Formula, "]" ) <> 0 Then 'Run Workbook
Link Remover and Cell Link Replacement
60 MacroLinkRemover
61 CalFormulaReplacement
62 End If
63 'GroupCal
64 On Error GoTo NoCal
65 If DayName = "Day" Then DayName = Day(Date) & "Day" 'Set Default Day
66 With ActiveSheet.Shapes(DayName)
67 .Fill.ForeColor.RGB = RGB(252, 213, 180)
68 .TextFrame2.TextRange.Font.Bold = msoTrue
69 End With
70 On Error GoTo NoCal
71 .Shapes("Calendar").Visible = msoCTrue
72 .Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3", "CalCol4",
"CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible = False '
73 .Shapes("Calendar").Left = SelCell.Left - 48
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",
"42Day")).Visible = True
79 Else:
80 .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
"42Day")).Visible = False
81 End If
82 Sheets("CalPopUp").Range("A7").Value = SelCell.Address
83 ActiveCell.Select
84 End With
85 Exit Sub
86 NoCal:
87 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
calendar over from another sheet and paste into this sheet"
88 End Sub
89 Sub CalFormulaReplacement()
90 With ActiveSheet
91 Dim DayNum, ColNum, RowNum As Long
92 Dim shp As Shape
93 ColNum = 2
94 RowNum = 1
95 For DayNum = 1 To 42
96 .Shapes(DayNum & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(RowNum
, ColNum).Address
97 ColNum = ColNum + 1
98 If ColNum = 9 Then
99 ColNum = 2
100 RowNum = RowNum + 1
101 End If
102 Next DayNum
103 .Shapes("Month").DrawingObject.Formula = "=CalPopUp!$A$4"
104 .Shapes("Year").DrawingObject.Formula = "=CalPopUp!$A$2"
105 End With
106 End Sub
107
108 Sub CheckForSheet()
109 'Checks for existance of Calendar Pop-up Worksheet
110 Dim ws As Worksheet
111 On Error GoTo CreateWS
112 Set ws = ActiveWorkbook.Sheets("CalPopUp")
113 Exit Sub

```

1

```

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

```

1 2

```

1 2
171 .Range("A20" ).Value =
    "=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) & ")"
172 'Set Defined Names
173 ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
174 ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
175
176 'Add in Calendar Formulas
177
178 .Range("B1" ).Value =
    "=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,Ca
lMonths,0),1)," & Chr(34) & Chr(34) & ")"
179 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
    ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
180 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
    ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
181 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
    ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
182 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
    ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
183 .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
    ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
184 .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
    ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
185 .Range("B2" ).Value = "=H1+1"
186 .Range("C2" ).Value = "=B2+1"
187 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
188 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
189 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
    & Chr(34) & Chr(34) & ",H4+1)"
190 .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",B5+1)," & Chr(34) & Chr(34) & ")"
191 .Range("B6" ).Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) &
    ",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) &
    ")"
192 .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",I5+1)," & Chr(34) & Chr(34) & ")"
193 .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
194
195 'Set format to Single Day
196 .Range("B1:H6" ).NumberFormat = "d"
197
198 'Add in relative Day #'s
199 .Range("B7" ).Value = "1"
200 .Range("C7" ).Value = "2"
201 .Range("B8" ).Value = "8"
202 .Range("C8" ).Value = "9"
203 .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
204 .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
205 GroupCal
206 End With
207 Exit Sub
208 NoCal:
209 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
    calendar over from another sheet and paste into this sheet"

```

```

1
210 End Sub
211 Sub GroupCal()
212     ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
213     Selection.ShapeRange.Name = "NextMonth"
214     ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
215     Selection.ShapeRange.Name = "PrevMonth"
216     ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
217         , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" ,
218         "1Day" , _
219         "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" ,
220         "6Day" , _
221         "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
222         "18Day" _
223         , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day"
224         , _
225         "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
226         "Sa" , _
227         "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
228         "CalCol3" , _
229         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" ,
230         "PrevMonth" , _
231         "NextMonth" , "NextYr" , "PrevYr" )).Visible = msoCTrue
232     ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
233         , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day"
234         , _
235         "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
236         "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
237         "18Day" _
238         , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
239         "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
240         "Sa" , _
241         "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
242         "CalCol3" , _
243         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth"
244         , _
245         "NextMonth" , "NextYr" , "PrevYr" )).Select
246     Selection.ShapeRange.Group.Select
247     Selection.ShapeRange.Name = "Calendar"
248     Selection.Name = "Calendar"
249     Selection.Placement = xlMove
250     ActiveSheet.Shapes("Calendar").Placement = 2
251 End Sub
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

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

```



```

1 2 3
314     .Range("A3" ).Value = 12
315     .Range("A2" ).Value = .Range("A2" ).Value - 1
316     Else:
317     .Range("A3" ).Value = .Range("A3" ).Value - 1
318     End If
319     If .Range("A6" ).Value > 0 Then
320     ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
    "40Day" , "41Day" , "42Day" )).Visible = True
321     Else:
322     ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
    "40Day" , "41Day" , "42Day" )).Visible = False
323     End If
324     End With
325     End Sub
326
327     Sub PrevYear()
328     ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
    "CalPopUp" ).Range("A2" ).Value - 1
329     End Sub
330     Sub ReplaceCalendar() 'Shape Deleted
331     ThisWorkbook.Sheets("CalPopUp" ).Shapes("Calendar" ).Copy 'Copy From Developers
    sheet
332     ActiveCell.Select 'Select the active Cell
333     ActiveSheet.Paste 'Paste in Calendar
334     End Sub
335
336     .....
337     ''''Select Day Of The Month
338     .....
339     Sub SelectDay()
340     Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
341     DayNumb = Replace(Application.Caller, "Day" , "" )
342     RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
343     ColNumb = DayNumb Mod 7 + 1
344     If ColNumb = 1 Then ColNumb = 8
345     'On Error Resume Next
346     If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
347     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
    ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
348     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
349     ActiveCell.Offset(0, 1).Select
350     End Sub
351     Sub ShowSettings()
352     'Show or Hide Calendar Settings Panel
353     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
354     ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
    "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
    = False
355     Else:
356     ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
    "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
    = True
357     End If
358     End Sub
359
360     Sub UngroupCal()
361     On Error Resume Next
362     ActiveSheet.Shapes("Calendar" ).Ungroup
363     ActiveSheet.Shapes("NextMonth" ).Ungroup
364     ActiveSheet.Shapes("PrevMonth" ).Ungroup
365     On Error GoTo 0

```

1

366

<sup>1</sup> End Sub

```

1 Option Explicit
2 Dim TaskShp As Shape
3 Dim LastRow As Long, ResultRow As Long, LastResultRow As Long
4 Dim ScRow As Long, ScStartCol As Long, ScEndCol As Long, StatusRow As Long
5 Dim TaskID As String, TaskName As String, EmpName As String, TaskStatus As String
6 Dim PicPath As String, EmpPicFolder As String, EmpPic As String, StatusColor As String
7 Dim ScInterv As Double
8 Dim ScDt As Date, StartDt As Date, EndDt As Date, StartTime As Date, EndTime As Date,
   ScStartTime As Date
9
10 Sub Sched_CheckForMove()
11     Dim CountDelay As Long, EmpName As String
12     Dim TaskStart As Double, TaskDur As Double, TaskRow As Long
13     With Main
14         If .Range("B9").Value = Empty Then Exit Sub
15         TaskID = .Range("B9").Value 'Task ID
16
17         For CountDelay = 1 To 100000
18             DoEvents
19             If .Range("B12").Value = True Then End 'Exit Loop on Move Task Move = True
20
21             With .Shapes("EmpGrp" & TaskID)
22                 If .Left <> Main.Range("B10").Value Or .Top <> Main.Range("B11").Value
23                     Then 'Move Detect
24                         'Check for incorrect move or missing Employee or time
25                         If .Left < Main.Range("A11").Left Or Main.Cells(4, .TopLeftCell.Column)
26                             .Value = "" Or .Top < Main.Range("A5").Top - 1 Or Main.Range("AH" & .
27                             TopLeftCell.Row).Value = "" Then
28                             MsgBox "Plase make sure to move the Task to a correct time on the
29                                 schedule"
30                             Schedule_Refresh
31                             .Range("B12").Value = True 'Set Move Task to True
32                             Exit Sub
33                         End If
34                         TaskRow = Main.Range("B13").Value ' Task Row
35                         EmpName = Main.Range("AH" & .TopLeftCell.Row).Value 'Employee Name
36                         TaskDur = Tasks.Range("I" & TaskRow).Value - Tasks.Range("G" & TaskRow
37                             ).Value
38                         TaskStart = Main.Cells(4, .TopLeftCell.Column).Value 'Start Time
39                         Tasks.Range("G" & TaskRow).Value = TaskStart 'Update Start Time
40                         Tasks.Range("I" & TaskRow).Value = TaskStart + TaskDur ' Update End
41                         time
42                         Tasks.Range("D" & TaskRow).Value = EmpName 'Update Employee Name
43                         Main.Range("B12").Value = True 'Set Move Appointment to True
44                         Schedule_Refresh 'Save Task Changes
45                     End If
46                 End With
47             Next CountDelay
48             .Range("B12").Value = True 'Set Move Task to True
49         End With
50     End Sub
51
52 Sub Sched_GoToTask()
53     With Main
54         If .Range("B9").Value = Empty Then Exit Sub
55         .Range("B4").Value = .Range("B9").Value
56         .Range("B12").Value = True 'Exit Check For Move Macro
57         .Range("Y5").Select 'Unselect shapes to reduce bugs
58         Menu_Change
59         Task_Load
60     End With
61

```

```

55 1 End Sub
56
57 Sub Sched_NextDay()
58     Main.Range("AI3").Value = Main.Range("AI3").Value + 1 'Set Next Day
59 End Sub
60
61 Sub Sched_PrevDay()
62     Main.Range("AI3").Value = Main.Range("AI3").Value - 1 'Set Previous Day
63 End Sub
64
65 Sub Sched_SelectTask()
66     With Main
67         TaskID = Replace(Application.Caller, Left(Application.Caller, 7), "" ) 'Extract
68         Task
69         .Range("B9").Value = TaskID
70         .Range("B10").Value = .Shapes("EmpGrp" & TaskID).Left
71         .Range("B11").Value = .Shapes("EmpGrp" & TaskID).Top
72         .Shapes("EmpGrp" & TaskID).Select
73         .Range("B12").Value = False 'Set Move to False
74         Sched_CheckForMove
75     End With
76 End Sub
77
78 Sub Sched_Today()
79     Main.Range("AI3").Value = Date 'Set current Date
80 End Sub
81 Sub Schedule_Refresh()
82     With Main
83         'Clear all Schedule Shapes
84         For Each TaskShp In .Shapes
85             On Error Resume Next
86             If InStr(TaskShp.Name, "EmpTask" ) > 0 Then TaskShp.Delete
87             If InStr(TaskShp.Name, "EmpPict" ) > 0 Then TaskShp.Delete
88             If InStr(TaskShp.Name, "EmpGrp" ) > 0 Then TaskShp.Delete
89             On Error GoTo 0
90         Next TaskShp
91         ScDt = .Range("AI3").Value 'Schedule Date
92         ScStartTime = Admin.Range("StartTime").Value 'Schedule Start Time
93         ScInterv = Admin.Range("Interval").Value 'Schedule Interval
94         EmpPicFolder = Admin.Range("PicFolder").Value 'Employee Picture folder
95     End With
96
97     With Tasks
98         .Range("AK3:AS9999").ClearContents 'Clear Prev. Results
99         LastRow = .Range("A99999").End(xlUp).Row
100        If LastRow < 4 Then Exit Sub
101        .Range("A3:R" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
102            "AH2:AI3" ), CopyToRange:=.Range("AK2:AR2" ), Unique:=True
103        LastResultRow = .Range("AK99999").End(xlUp).Row
104        If LastResultRow < 3 Then Exit Sub
105        .Range("AS3:AS" & LastResultRow).Formula = .Range("AS1").Formula 'Employee
106        Pictures
107        For ResultRow = 3 To LastResultRow
108            'Get Variables
109            TaskID = .Range("AK" & ResultRow).Value 'Task ID
110            TaskName = .Range("AL" & ResultRow).Value 'Task Name
111            EmpName = .Range("AM" & ResultRow).Value 'Employee Name
112            StartDt = .Range("AN" & ResultRow).Value 'Start Date
113            StartTime = .Range("AO" & ResultRow).Value 'StartTime
114            EndDt = .Range("AP" & ResultRow).Value 'End Date
115            EndTime = .Range("AQ" & ResultRow).Value 'End time

```

1 2 3

```

1 2 3
113 TaskStatus = .Range("AR" & ResultRow).Value 'Task Status
114 EmpPic = .Range("AS" & ResultRow).Value 'Employee Picture
115
116 'Set Starting Column
117 If StartDt < ScDt Or StartTime <= ScStartTime Then 'Set To first Column
118     ScStartCol = 35
119 Else 'Starts Within the day
120     ScStartCol = 35 + Int((StartTime - ScStartTime) / ScInterv) 'Set Start
121     Column
122 End If
123
124 'Set Ending Column
125 If EndDt > ScDt Or EndTime >= Main.Range("EndTime" ).Value Then
126     ScEndCol = 63
127 Else 'Ends Within the day
128     ScEndCol = 35 + Int((EndTime - ScStartTime) / ScInterv) 'Set End Column
129 End If
130
131 'Determine Task Status Color
132 On Error Resume Next
133 StatusRow = Admin.Range("Status" ).Find(TaskStatus, , xlValues, xlWhole).Row
134 On Error GoTo 0
135 If StatusRow <> 0 Then StatusColor = Admin.Range("O" & StatusRow).Interior.
136     Color 'Get Interior of cell color
137
138 'Determine Scheduld Row
139 On Error Resume Next
140 ScRow = Main.Range("AH5:AH999" ).Find(EmpName, , xlValues, xlWhole).Row 'Get
141     Employee Schedule Row
142 On Error GoTo 0
143 If ScRow = 0 Then GoTo NextTask
144
145 With Main
146     'Create Task Shape
147     .Shapes("SampleTaskShp" ).Duplicate.Name = "EmpTask" & TaskID
148     With .Shapes("EmpTask" & TaskID)
149         If StatusColor <> "" Then .Fill.ForeColor.RGB = StatusColor 'Set Shape
150         Task Color
151         .Left = Main.Cells(ScRow, ScStartCol).Left
152         .Top = Main.Cells(ScRow, ScStartCol).Top
153         .Width = Range(Main.Cells(ScRow, ScStartCol), Main.Cells(ScRow, ScEndCol
154         )).Width
155         .TextFrame2.TextRange.Text = TaskName 'Set Task Name
156     End With
157
158     'Set Employee or Default Picture
159     .Shapes("SampleEmplShp" ).Duplicate.Name = "EmpPict" & TaskID
160     PicPath = EmpPicFolder & "\" & EmpPic
161     With .Shapes("EmpPict" & TaskID)
162         .Left = Main.Cells(ScRow, ScStartCol).Left
163         .Top = Main.Cells(ScRow, ScStartCol).Top
164         If Dir(PicPath, vbDirectory) <> "" Then .Fill.UserPicture (PicPath)
165     End With
166 End With
167
168 With Main.Shapes.Range(Array("EmpTask" & TaskID, "EmpPict" & TaskID)).Group
169     .Name = "EmpGrp" & TaskID
170 End With
171 NextTask:
172 Next ResultRow
173 End With
174 End Sub

```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, TaskRow As Long, TaskCol As Long, SelRow As
  Long
3 Sub Task_Delete()
4   With Main
5     If MsgBox("Are you sure you want to delete this Task?" , vbYesNo, "Delete Task" )
6       = vbNo Then Exit Sub
7     If .Range("B5" ).Value = Empty Then GoTo NotSaved
8     TaskRow = .Range("B5" ).Value 'Task Row
9     Tasks.Range(TaskRow & ":" & TaskRow).EntireRow.Delete 'Delete Row
10    NotSaved:
11    Task_New 'Run macro to create new Task
12    Task_ListTasks 'Update Task list
13  End With
14 End Sub
15
16 Sub Task_ListTasks()
17   Main.Range("B3,Q6:V9999" ).ClearContents 'Clear existing Tasks
18   With Tasks
19     LastRow = .Range("A9999" ).End(xlUp).Row 'Last Row of data
20     If LastRow < 4 Then Exit Sub
21     .Range("A3:R" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
22       "W2:W3" ), CopyToRange:=.Range("Z2:AE2" ), Unique:=True
23     LastResultRow = .Range("Z99999" ).End(xlUp).Row 'Last Result Row
24     If LastResultRow < 3 Then Exit Sub
25     If LastResultRow < 4 Then GoTo SkipSort
26     With .Sort
27       .SortFields.Clear
28       .SortFields.Add Key:=Tasks.Range("Z3" ), SortOn:=xlSortOnValues, Order:=
29         xlAscending, DataOption:=xlSortNormal 'Sort
30     .SetRange Tasks.Range("Z3:AE" & LastResultRow) 'Set Range
31     .Apply 'Apply Sort
32   End With
33   SkipSort:
34   Main.Range("Q6:V" & LastResultRow + 3).Value = .Range("Z3:AE" & LastResultRow).
35   Value 'Bring Over Tasks
36   On Error Resume Next
37   SelRow = Main.Range("Q6:Q9999" ).Find(Main.Range("B4" ).Value, , xlFormulas,
38     xlWhole).Row
39   If SelRow <> 0 And Main.Range("B4" ).Value <> Empty Then Main.Range("B3" ).Value
40   = SelRow 'Set Selected Row
41   On Error GoTo 0
42 End With
43 End Sub
44
45 Sub Task_Load()
46   With Main
47     If .Range("B5" ).Value = Empty Then
48       MsgBox "Please select a correct Task to load"
49       Exit Sub
50     End If
51     TaskRow = .Range("B5" ).Value 'Task Row
52     For TaskCol = 2 To 15
53       .Range(Tasks.Cells(1, TaskCol).Value).Value = Tasks.Cells(TaskRow, TaskCol).
54       Value 'Bring over data
55     Next TaskCol
56   End With
57 End Sub
58
59 Sub Task_MakeRecurring()

```

```

1
54 Dim Freq As String
55 Dim FreqQty As Long
56 Dim TotTime As Double
57 Dim StartOnDt As Date, Untildt As Date, StartDt As Date, EndDt As Date
58 With Main
59     If .Range("Y5" ).Value = Empty Then
60         MsgBox "Please make sure to enter a Task Name before saving"
61         Exit Sub
62     End If
63     If .Range("Y7" ).Value = Empty Or .Range("Y9" ).Value = Empty Then
64         MsgBox "Please make sure task has Start and End Dates to make them recurring"
65
66         Exit Sub
67     End If
68     If .Range("B7" ).Value < 4 Then
69         MsgBox "Please make sure to enter Recurring Frequency, Start On and Until
70         Date fields to make this task Recurring"
71         Exit Sub
72     End If
73     TotTime = .Range("AE9" ).Value / 24 'Total Time
74     FreqQty = .Range("Y18" ).Value 'Frequency Qty
75     Freq = .Range("Z18" ).Value 'Frequency
76     StartOnDt = .Range("AB18" ).Value 'Start On Date
77     Untildt = .Range("AD18" ).Value 'Until Date
78
79     StartDt = StartOnDt 'Set Initial Starting Date
80     EndDt = StartDt + TotTime 'End Date is Start Date + Total Time
81
82     Do While StartDt <= Untildt 'Create Tasks Until Start Date is greater than Until
83     Date
84         .Range("Y7" ).Value = StartDt 'Set Starting Date
85         .Range("Y9" ).Value = Int(EndDt) 'Set Ending Date (date as a whole number)
86         .Range("B4" ).ClearContents 'Clear Existing Task ID to Create New Task
87         Task_SaveUpdate 'Save Task
88
89         'Update Start & End Dates for Next Task
90         Select Case Freq
91         Case Is = "Day(s)"
92             StartDt = DateAdd("d" , FreqQty, StartDt)
93         Case Is = "Weekday(s)"
94             StartDt = DateAdd("w" , FreqQty, StartDt)
95             If Weekday(StartDt) = 7 Then StartDt = StartDt + 2
96             If Weekday(StartDt) = 1 Then StartDt = StartDt + 1
97         Case Is = "Week(s)"
98             StartDt = DateAdd("ww" , FreqQty, StartDt)
99         Case Is = "Months(s)"
100             StartDt = DateAdd("m" , FreqQty, StartDt)
101         End Select
102         EndDt = StartDt + TotTime 'Update End Date
103     Loop
104 End With
105 Task_ListTasks 'Update Task list
106 End Sub
107
108 Sub Task_New()
109     Main.Range("B3,B4,Y5,AB5,AE5,Y7,AB7,AE7,Y9,AB9,Y11,AB11,AE11,Y13,AB13,Y15:AE15" ).
110     ClearContents 'Clear Task Fields
111 End Sub
112
113 Sub Task_SaveUpdate()
114     With Main
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```
1 2
111 If .Range("Y5").Value = "" Then 'Task Name
112     MsgBox "Please make sure to enter a Task Name before saving"
113     Exit Sub
114 End If
115 If .Range("AE5").Value = "" Then
116     MsgBox "Please make sure to enter an Employee Name before saving"
117     Exit Sub
118 End If
119 If .Range("B5").Value = Empty Then 'New Task
120     TaskRow = Tasks.Range("A99999").End(xlUp).Row + 1 'First Avail Row
121     .Range("B4").Value = .Range("B6").Value 'New Task. ID
122     Tasks.Range("A" & TaskRow).Value = .Range("B4").Value 'Set Task ID
123     Tasks.Range("R" & TaskRow).Value = "=Row()" 'Set Row
124 Else 'Existing Task
125     TaskRow = .Range("B5").Value 'Existing Task Row
126 End If
127 For TaskCol = 2 To 17
128     Tasks.Cells(TaskRow, TaskCol).Value = .Range(Tasks.Cells(1, TaskCol).Value).
129     Value 'Save/Update Data
130 Next TaskCol
131 If Application.Caller = "SaveBtn" Then
132     MsgBox "Task has been saved"
133     Task_ListTasks 'Update Task list
134 End If
135 End With
136 End Sub
```



- 
- , 19
- A**
- Activate, 17
- ActiveCell, 11, 15, 16, 21
- ActiveSheet, 15-17, 19-21
- ActiveWorkbook, 16-18
- ActSht, 17
- Add, 6, 13, 14, 17, 18, 26
- Address, 16, 17
- Admin, 11, 12, 24, 25
- Admin\_SetStatusColor, 11
- AdvancedFilter, 12, 13, 24, 26
- After, 17
- AllowMultiSelect, 11, 14
- Application, 11, 13-15, 21, 24, 28
- Apply, 13, 26
- Array, 15, 16, 19-21, 25
- AttFile, 14
- AutoFill, 17, 18
- B**
- Bold, 15, 16, 20
- BrowseForEmplPicFolder, 11
- BtnNumb, 13, 14
- Button, 6
- C**
- Cal, 7
- CalCol, 15, 17
- CalendarHide, 7, 15
- CalendarShow, 7, 15
- CalFormulaReplacement, 16
- Caller, 11, 13, 15, 21, 24, 28
- Cells, 16, 17, 21, 23, 25, 26, 28
- CheckForSheet, 7, 16
- Chr, 17, 18
- Clear, 13, 26
- ClearContents, 12, 13, 24, 26, 27
- ColCnt, 17
- ColNum, 16
- ColNumb, 21
- Color, 11, 25
- Column, 23
- Copy, 17, 21
- CopyToRange, 12, 13, 24, 26
- Count, 17
- CountDelay, 23
- CountLarge, 3, 6
- CreateCalSht, 17
- CreateWS, 16, 17
- CriteriaRange, 12, 24, 26
- D**
- Dashboard\_Refresh, 6, 12
- DashData, 12
- DataOption, 13, 26
- DateAdd, 27
- DateRange\_Change, 6, 12
- Day, 16
- DayCnt, 17
- DayName, 15, 16, 20
- DayNum, 15, 16
- DayNumb, 21
- Delete, 6, 24, 26
- Destination, 17, 18
- Dir, 25
- DoEvents, 23
- DrawingObject, 16, 17
- Duplicate, 25
- E**
- Employee\_Sort, 13
- Employees, 13
- EmpName, 23-25
- EmpPic, 23, 25
- EmpPicFolder, 23-25
- Empty, 3, 6, 7, 15, 20, 21, 23, 26-28
- EndDt, 23-25, 27
- EndTime, 23-25
- EntireColumn, 13
- EntireRow, 26
- Explicit, 3-6, 8-13, 23, 26
- F**
- FileDialog, 11, 14
- Fill, 6, 11, 13-16, 20, 25
- Filters, 14
- Find, 12, 25, 26
- Font, 6, 13-16, 20
- ForeColor, 6, 11, 13-16, 20, 25
- Formula, 16, 17, 24
- Freq, 27
- FreqQty, 27
- G**
- Group, 19, 25
- GroupCal, 18, 19
- H**
- Hidden, 13
- Hover1\_MouseMove, 6
- Hover2\_MouseMove, 6
- Hover3\_MouseMove, 6
- Hover4\_MouseMove, 6
- I**
- InStr, 16, 20, 24
- Int, 25, 27
- Interior, 11, 25
- Intersect, 3, 6, 7
- IsDate, 15
- J**
- JobFolder, 11
- K**
- Key, 13, 26
- L**
- LastResultRow, 12, 23, 24, 26
- LastRow, 12, 13, 23, 24, 26
- Left, 3, 13, 16, 20, 23-25
- Len, 20
- M**
- MacroLink, 19, 20
- MacroLinkRemover, 16, 19
- Main, 6, 12-14, 23-27
- Menu\_Change, 13, 23
- Menu\_ColorDefault, 6, 13
- MenuNumb, 13
- Month, 15, 17
- MsgBox, 12, 16, 18, 23, 26-28
- msoBringToFront, 16
- msoCTrue, 3, 13, 16, 19
- msoFalse, 3, 11, 13, 15, 20, 21
- msoFileDialogFilePicker, 14
- msoFileDialogFolderPicker, 11
- msoTrue, 16
- N**
- Name, 17-19, 24, 25
- Names, 18
- NewLink, 19, 20
- NextMonth, 20
- NextShp, 19, 20
- NextTask, 25
- NextYear, 20
- NoCal, 15, 16, 18
- NoSelection, 11, 14
- NotSaved, 26
- NumberFormat, 18
- O**
- Offset, 16, 21
- OnAction, 17, 19, 20
- Order, 13, 26
- P**
- Paste, 17, 21
- PicPath, 23, 25
- Placement, 16, 19
- PrevMonth, 20
- PrevYear, 21
- R**
- Range, 3, 6, 7, 11-21, 23-28
- RangeRow, 12
- RefersTo, 18
- Replace, 13, 21, 24
- ReplaceCalendar, 7, 21
- ResultRow, 12, 23-25
- RGB, 6, 11, 13-16, 20, 25
- Right, 20
- RoundUp, 21
- Row, 3, 7, 12, 13, 23-26, 28
- RowCnt, 17
- RowNum, 16
- RowNumb, 21
- S**
- ScDt, 23-25
- ScEndCol, 23, 25
- Sched\_CheckForMove, 23, 24
- Sched\_GoToTask, 23
- Sched\_NextDay, 24
- Sched\_PrevDay, 24
- Sched\_SelectTask, 24
- Sched\_Today, 24
- Schedule\_Refresh, 6, 23, 24
- ScInterv, 23-25
- ScRow, 23, 25
- ScStartCol, 23, 25
- ScStartTime, 23-25
- SelCell, 15, 16
- SelectDay, 21
- SelectedItems, 11, 14
- Selection, 15, 19
- SelRow, 26
- SetRange, 13, 26
- Shape, 7, 16, 19, 23
- ShapeRange, 15, 19
- Shapes, 3, 6, 7, 11, 13-17, 19-21, 23-25
- Sheets, 15-17, 20, 21
- Shift, 6
- Show, 11, 14
- ShowSettings, 21

shp, 16, 19, 20  
SkipSort, 26  
Sort, 13, 26  
SortFields, 13, 26  
SortOn, 13, 26  
Split, 20  
SplitLink, 19, 20  
StartDt, 23-25, 27  
StartOnDt, 27  
StartTime, 23-25  
StatusColor, 23, 25  
StatusRow, 23, 25

## T

Target, 3, 6, 7  
Task\_Delete, 26  
Task\_ListTasks, 6, 26-28  
Task\_Load, 7, 23, 26  
Task\_MakeRecurring, 26  
Task\_New, 26, 27  
Task\_SaveUpdate, 27  
TaskCol, 26, 28  
TaskDur, 23  
TaskID, 23-25  
TaskName, 23-25  
TaskRow, 23, 26, 28  
Tasks, 12, 13, 23, 24, 26, 28  
Tasks\_AddAttach, 14  
TaskShp, 23, 24  
TaskStart, 23  
TaskStatus, 23, 25  
Text, 25  
TextFrame2, 6, 13-16, 20, 25  
TextRange, 6, 13-16, 20, 25  
ThisWorkbook, 17, 20, 21  
Title, 11, 14  
Top, 3, 13, 16, 23-25  
TopLeftCell, 23  
TotTime, 27

## U

Ungroup, 21  
UnGroupCal, 17, 21  
Unique, 12, 13, 24, 26  
UntilDt, 27  
UserPicture, 25

## V

Validation, 6  
Value, 3, 6, 7, 11-18, 20, 21, 23-28  
vbDirectory, 25  
vbNo, 26  
vbYesNo, 26  
Visible, 3, 7, 11, 13, 15-17, 19-21

## W

Weekday, 27  
Width, 25  
Worksheet, 16, 17  
Worksheet\_Change, 6  
Worksheet\_SelectionChange, 3, 6  
WorksheetFunction, 21  
ws, 16, 17

## X

X, 6  
xlAscending, 13, 26  
xlFillDefault, 17, 18  
xlFilterCopy, 12, 13, 24, 26  
xlFormulas, 26

xlMove, 16, 19  
xlSheetHidden, 17  
xlSortNormal, 13, 26  
xlSortOnValues, 13, 26  
xlUp, 12, 13, 24, 26, 28  
xlValidateList, 6  
xlValues, 12, 25  
xlWhole, 12, 25, 26

## Y

Y, 6  
Year, 15, 17

## Z

ZOrder, 16

# 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*