

VBA SOURCE CODE BOOK

FIELD SERVICE MANAGER UPDATED



EXCLUSIVE FOR
SILVER & GOLD
MEMBERS

FEATURE, FIX, OR FOCUS



Excel For Freelancers

The Best Field Service Application Build With Excel



[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 300,000 YouTube subscribers, 15,000,000 video views, 200+ comprehensive training videos, and a thriving 50,000 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



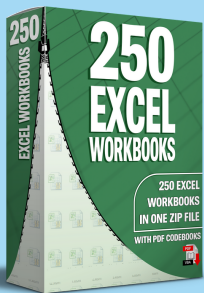
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



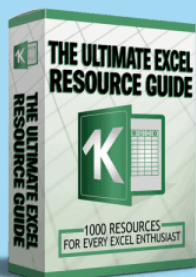
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



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](#)

Table of Contents

Projects.....	3
VBAProject.....	3
Documents.....	3
Admin.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub].....	3
CalPopUp.....	4
(Declarations).....	4
Customers.....	5
(Declarations).....	5
Dashboard.....	6
(Declarations).....	6
Home.....	7
(Declarations).....	7
Items.....	8
(Declarations).....	8
JobMap.....	9
(Declarations).....	9
Worksheet_Change [Sub].....	9
Worksheet_SelectionChange [Sub].....	9
OrderItemDB.....	10
(Declarations).....	10
Orders.....	11
(Declarations).....	11
Worksheet_Change [Sub].....	11
Worksheet_SelectionChange [Sub].....	11
OrdersDB.....	13
(Declarations).....	13
Technicians.....	14
(Declarations).....	14
ThisWorkbook.....	15
(Declarations).....	15
Modules.....	16
App_Macros.....	16
(Declarations).....	16
Admin_SetStatusColor [Sub].....	16
BrowseForMapFolder [Sub].....	16
BrowseForTechFolder [Sub].....	16
Menu_GoHome [Sub].....	16
Menu_GoToSheet [Sub].....	16
Map_Macros.....	17
(Declarations).....	17
Map_CloseEdit [Sub].....	17
Map_EditJob [Sub].....	17
Map_NextDay [Sub].....	17
Map_PrevDay [Sub].....	17
Map_RefreshJobs [Sub].....	17
Map_SaveAsCustLoc [Sub].....	19
Map_SaveLocation [Sub].....	19
Map_SelectOrder [Sub].....	19
Map_ShowRoute [Sub].....	20
Map_Today [Sub].....	21
Order_Macros.....	22
(Declarations).....	22
Cust_SavedMsg [Sub].....	22
Customer_SaveUpdate [Sub].....	22
Order_Delete [Sub].....	22
Order_Load [Sub].....	23
Order_New [Sub].....	23
Order_Print [Sub].....	24
Order_SavedMsg [Sub].....	24
Order_SaveUpdate [Sub].....	24
Orders_ClearFilter [Sub].....	25
Orders_LoadList [Sub].....	25
PopUp_Calendar.....	26
(Declarations).....	26

Table of Contents

CalCol [Sub]	26
CalendarHide [Sub].....	26
CalendarShow [Sub].....	26
CalFormulaReplacement [Sub]	27
CheckForSheet [Sub].....	27
CreateCalSht [Sub]	28
GroupCal [Sub]	30
MacroLinkRemover [Sub]	30
NextMonth [Sub]	31
NextYear [Sub]	31
PrevMonth [Sub]	31
PrevYear [Sub]	32
ReplaceCalendar [Sub].....	32
SelectDay [Sub]	32
ShowSettings [Sub]	32
UnGroupCal [Sub]	32

```
1 Option Explicit
2
3
4 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
5     If Target.CountLarge > 1 Then Exit Sub
6     If Shapes("ColorPalette" ).Visible = True Then Shapes("ColorPalette" ).Visible =
7         msoFalse
8
9     'On Work Day Change
10    If Not Intersect(Target, Range("D17:D23" )) Is Nothing Then
11        If Target.Value = Empty Then Target.Value = Chr(252) Else Target.ClearContents
12        Range("D8" ).Select
13    End If
14
15    'Show color palette
16    If Not Intersect(Target, Range("D9:D13" )) Is Nothing Then
17        With Shapes("ColorPalette" )
18            .Left = Range("D" & Target.Row).Left
19            .Top = Range("D" & Target.Row + 1).Top
20            .Visible = msoCTrue
21        End With
22    End If
23 End Sub
```

1
2

Option Explicit

1 Option Explicit

2

1
2

Option Explicit

1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("S2,E2,AV2" )) Is Nothing Then Map_RefreshJobs
5 End Sub
6
7 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
8     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
9     it in
10    Dim Cal As Shape
11    On Error Resume Next
12    Set Cal = Shapes("Calendar" )
13    On Error GoTo 0
14    If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
15    If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
16
17    If Not Intersect(Target, Range("E2" )) Is Nothing Then 'Date Range Selection
18        CheckForSheet
19        CalendarShow
20    Else:
21        CheckForSheet
22        CalendarHide
23    End If
24 End Sub
```

1	Option Explicit
2	

```
1 Option Explicit
```

```
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
```

```
4     If Target.CountLarge > 1 Then Exit Sub
```

```
5
6     'On Search By Change
```

```
7     If Not Intersect(Target, Range("E5" )) Is Nothing Then Orders_LoadList 'Run Macro
8     to Load list on Cell change
```

```
9
10    'Load Customer on Name change
```

```
11    If Not Intersect(Target, Range("H4" )) Is Nothing Then
```

```
12
13        If Range("B9" ).Value <> "" Then 'Existing Customer
```

```
14            Dim CustRow As Long
```

```
15            CustRow = Range("B9" ).Value 'Customer Row
```

```
16            Range("H6" ).Value = Customers.Range("C" & CustRow).Value 'Customer Name
```

```
17            Range("H7" ).Value = Customers.Range("D" & CustRow).Value 'Customer Address
```

```
18            Range("H8" ).Value = Customers.Range("E" & CustRow).Value 'Customer City
```

```
19            Range("J8" ).Value = Customers.Range("F" & CustRow).Value 'Customer State
```

```
20            Range("L8" ).Value = Customers.Range("G" & CustRow).Value 'Customer Zip
```

```
21            Range("L9" ).Value = Customers.Range("H" & CustRow).Value 'Customer Phone
```

```
22            Range("H9" ).Value = Customers.Range("I" & CustRow).Value 'Customer Email
```

```
23            Range("B12" ).Value = Customers.Range("J" & CustRow).Value 'Left Location
24            (if any)
```

```
25            Range("B13" ).Value = Customers.Range("K" & CustRow).Value 'Top Location (if
26            any)
```

```
27        Else 'No Customer (Clear Fields)
```

```
28            Range("H6,H7:L7,H8,J8,L8,H9:J9,L9,B12,B13" ).ClearContents
```

```
29        End If
```

```
30    End If
```

```
31    'On Item Change but not on Work Order Load
```

```
32    If Not Intersect(Target, Range("G18:G66" )) Is Nothing And Range("B8" ).Value =
33    False Then
```

```
34        If Range("G" & Target.Row).Value <> Empty Then
```

```
35            Dim ItemDBRow As Long
```

```
36            On Error Resume Next
```

```
37            ItemDBRow = Items.Range("Item_Name" ).Find(Target.Value, , xlValues, xlWhole).
38            Row
```

```
39            On Error GoTo 0
```

```
40            If ItemDBRow = 0 Then
```

```
41                MsgBox "Please select a correct Item from the drop down list"
```

```
42                Exit Sub
```

```
43            End If
```

```
44            'Add in Item details
```

```
45            Range("H" & Target.Row).Value = Items.Range("E" & ItemDBRow).Value 'Item
46            Description
```

```
47            Range("J" & Target.Row).Value = Items.Range("F" & ItemDBRow).Value 'Default
48            Qty/Hours
```

```
49        Else 'Empty, Clear out Row
```

```
50            Range("B" & Target.Row & ":L" & Target.Row).ClearContents
```

```
51        End If
```

```
52    End If
```

```
53 End Sub
```

```
54
55 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
56     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
57     it in
```

```
1
```

```
1
54 Dim Cal As Shape
55 On Error Resume Next
56 Set Cal = Shapes("Calendar" )
57 On Error GoTo 0
58 If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
59 If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
60
61 If Target.CountLarge > 1 Then Exit Sub
62
63 'On selection of Orders
64 If Not Intersect(Target, Range("D7:E9999" )) Is Nothing And Range("D" & Target.Row)
65 .Value <> Empty Then
66     Range("B3" ).Value = Range("D" & Target.Row).Value 'Set Order ID
67     Range("B7" ).Value = Target.Row 'Set Selected Row
68     Order_Load
69 End If
70
71 If Not Intersect(Target, Range("H11" )) Is Nothing Then 'Change this to any cell
72 (or cells) you would like to have the Pop-Up Calendar Appear
73     CheckForSheet
74     CalendarShow
75 Else:
76     CheckForSheet
77     CalendarHide
78 End If
79
80 End Sub
```

1
2

Option Explicit

1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2 Sub Admin_SetStatusColor()
3     ActiveCell.Interior.Color = Admin.Shapes(Application.Caller).Fill.ForeColor.RGB
4     Admin.Shapes("ColorPalette").Visible = msoFalse
5 End Sub
6
7
8 Sub BrowseForMapFolder()
9     Dim JobFolder As FileDialog
10    Set JobFolder = Application.FileDialog(msoFileDialogFolderPicker)
11    With JobFolder
12        .Title = "Browse For Map Folder"
13        .AllowMultiSelect = False
14        If .Show <> -1 Then GoTo NoSelection
15        Admin.Range("E6").Value = .SelectedItems(1)
16    End With
17    NoSelection:
18 End Sub
19
20 Sub BrowseForTechFolder()
21     Dim JobFolder As FileDialog
22     Set JobFolder = Application.FileDialog(msoFileDialogFolderPicker)
23     With JobFolder
24         .Title = "Browse For Technician Folder"
25         .AllowMultiSelect = False
26         If .Show <> -1 Then GoTo NoSelection
27         Admin.Range("E5").Value = .SelectedItems(1)
28     End With
29     NoSelection:
30 End Sub
31 Sub Menu_GoHome()
32     Home.Activate
33 End Sub
34
35 Sub Menu_GoToSheet()
36     Sheets(Application.Caller).Activate
37     Application.ScreenUpdating = True
38 End Sub
```

```

1 Option Explicit
2 Dim JobNm As String, OrderID As String, PicPath As String, PicFolder As String, Status
  As String, StatusColor As String, TechNm As String, NextOrderID As String
3 Dim OrderRow As Long, LastRow As Long, LastResultRow As Long, ResultRow As Long,
  StatusRow As Long, MoveCar As Long
4 Dim LeftPos As Double, TopPos As Double, UnMapTop As Double, StartLeft As Double,
  StartTop As Double, NextLeft As Double, NextTop As Double
5 Dim JobShp As Shape
6 Dim MapPic As String, MapFolder As String, MapPath As String
7
8 Sub Map_CloseEdit()
9     With JobMap
10        OrderID = Replace(Application.Caller, "CloseJob" , "" )
11        .Shapes("SelJobGrp" & OrderID).Ungroup 'Ungroup Selected Group
12        .Shapes("EditJobGrp" & OrderID).Delete 'Delete edit group
13    End With
14 End Sub
15
16 Sub Map_EditJob()
17     With JobMap
18        OrderID = Replace(Application.Caller, "EditJob" , "" )
19        .Shapes("SelJobGrp" & OrderID).Ungroup 'Ungroup Selected Group
20        .Shapes("EditJobGrp" & OrderID).Delete 'Delete edit group
21    End With
22    Orders.Range("B3" ).Value = OrderID
23    Orders.Activate
24    Order_Load 'Run Macro to Load Order
25
26 End Sub
27
28 Sub Map_NextDay()
29     JobMap.Range("E2" ).Value = JobMap.Range("E2" ).Value + 1 'Next Day
30 End Sub
31
32 Sub Map_PrevDay()
33     JobMap.Range("E2" ).Value = JobMap.Range("E2" ).Value - 1 'Previous Day
34 End Sub
35
36 Sub Map_RefreshJobs()
37     'Clear All Map jobs
38     PicFolder = [TecFolder] 'Set Technician Picture Folder
39     With JobMap
40         For Each JobShp In .Shapes
41             On Error Resume Next
42             If InStr(JobShp.Name, "JobGrp" ) > 0 Then JobShp.Delete
43             On Error GoTo 0
44         Next JobShp
45     End With
46     With OrdersDB
47         .Range("AA3:AH999" ).ClearContents 'Clear Previous results
48         LastRow = .Range("A99999" ).End(xlUp).Row
49         If LastRow < 4 Then Exit Sub
50         .Range("M4:M" & LastRow).Formula = .Range("M2" ).Formula 'Technician Type
         Formula
51         .Range("A3:M" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
         "V2:X3" ), CopyToRange:=.Range("AA2:AG2" ), Unique:=True
52         LastResultRow = .Range("AA99999" ).End(xlUp).Row
53         If LastResultRow < 3 Then GoTo NoResults
54         .Range("AH3:AH" & LastResultRow).Formula = .Range("AH1" ).Formula 'Copy Down
         Tech Pic. File Name
55     NoResults:

```

1 2

```

56      1 2
57      UnMapTop = JobMap.Range("A6" ).Top 'Set initial unmapped top position
58
59      'Set Map Background Picture
60      MapPic = .Range("Y3" ).Value 'Set Map Picture
61      MapFolder = Admin.Range("E6" ).Value 'Map Folder
62      MapPath = MapFolder & "\" & MapPic 'Full File Path Map
63      If Dir(MapPath, vbDirectory) <> "" Then JobMap.SetBackgroundPicture MapPath
64
65      If LastResultRow < 3 Then Exit Sub
66      For ResultRow = 3 To LastResultRow
67          'Get Job Data
68          OrderID = .Range("AA" & ResultRow).Value 'Set Order ID
69          JobNm = .Range("AB" & ResultRow).Value 'Job Name
70          Status = .Range("AC" & ResultRow).Value 'Job Status
71          On Error Resume Next
72          StatusRow = Admin.Range("Status" ).Find(Status, , xlValues, xlWhole).Row 'Set
          Status Row
73          On Error GoTo 0
74          If StatusRow <> 0 Then StatusColor = Admin.Range("D" & StatusRow).Interior.
          Color 'Set Status Color
75          LeftPos = .Range("AD" & ResultRow).Value 'Left Location
76          TopPos = .Range("AE" & ResultRow).Value 'Top Location
77          If LeftPos = 0 Or TopPos = 0 Then 'Set Unmapped position
78              LeftPos = .Range("A1" ).Left
79              TopPos = UnMapTop 'Set Unmapped top position
80              UnMapTop = UnMapTop + 25 'Increment Unmapped top position
81          End If
82          PicPath = PicFolder & "\" & .Range("AH" & ResultRow).Value 'Tech Picture
          Path (if any)
83          With JobMap
84              'Create Job Shapes
85              'Set Map Icon
86              .Shapes("MapIconSample" ).Duplicate.Name = "MapIcon" & OrderID 'Duplicate
          Map Icon
87              .Shapes("MapIcon" & OrderID).Left = LeftPos 'Set Map Icon Left Position
88              .Shapes("MapIcon" & OrderID).Top = TopPos 'Set Map Icon Top Position
89
90              'Set Job Name Shape
91              .Shapes("JobNameSample" ).Duplicate.Name = "JobName" & OrderID
92              With .Shapes("JobName" & OrderID)
93                  .TextFrame2.TextRange.Text = JobNm 'Set Job Name
94                  .TextFrame2.AutoSize = msoAutoSizeShapeToFitText 'Dynamic width based
          on Job Name Length
95                  .Height = 15 'Set Fixed Height
96                  .Left = LeftPos + 11 'Set Left Position
97                  .Top = TopPos + 17 'Set Top Position
98                  If StatusColor <> "" Then .Line.ForeColor.RGB = StatusColor 'Set Line
          Color Based On Status
99              End With
100
101              'Set Tech Icon
102              .Shapes("TechPicSample" ).Duplicate.Name = "TechPic" & OrderID 'Duplicate
          Tech Icon
103              With .Shapes("TechPic" & OrderID)
104                  If Dir(PicPath, vbDirectory) <> "" And OrdersDB.Range("AH" & ResultRow
          ).Value <> "" Then .Fill.UserPicture (PicPath) 'Add User Picture (if
          available)
105                  .Left = LeftPos + 1.75
106                  .Top = TopPos + 1
107              End With

```

```

1 2 3 4
108 .Shapes.Range(Array("MapIcon" & OrderID, "JobName" & OrderID, "TechPic"
    & OrderID)).Group.Name = "JobGrp" & OrderID
109 End With
110 StatusColor = ""
111 Next ResultRow
112 End With
113
114 End Sub
115
116 Sub Map_SaveAsCustLoc()
117 Dim CustRow As Long
118 With JobMap
119     If .Range("BM6" ).Value = Empty Then
120         MsgBox "Please select a job before saving this location"
121         Exit Sub
122     End If
123     CustRow = .Range("BM6" ).Value 'Set Customer Row
124     OrderID = .Range("BM4" ).Value 'Order ID
125     On Error Resume Next
126     Set JobShp = .Shapes("SelJobGrp" & OrderID)
127     On Error GoTo 0
128     If JobShp Is Nothing Then
129         MsgBox "Please select a job before saving this location"
130         Exit Sub
131     End If
132     Customers.Range("J" & CustRow).Value = JobShp.Left 'Save Left Position
133     Customers.Range("K" & CustRow).Value = JobShp.Top 'Save Top Position
134     MsgBox "Customer Location Saved"
135 End With
136
137 End Sub
138
139
140 Sub Map_SaveLocation()
141 With JobMap
142     OrderID = Replace(Application.Caller, "SaveJob" , "" )
143     .Range("BM4" ).Value = OrderID 'Order ID
144     OrderRow = .Range("BM5" ).Value 'Order Row
145     OrdersDB.Range("K" & OrderRow).Value = .Shapes("SelJobGrp" & OrderID).Left
    'Set Left Postion
146     OrdersDB.Range("L" & OrderRow).Value = .Shapes("SelJobGrp" & OrderID).Top 'Set
    top Postion
147     .Shapes("SelJobGrp" & OrderID).Ungroup 'Ungroup Selected Group
148     .Shapes("EditJobGrp" & OrderID).Delete 'Delete edit group
149 End With
150 Map_RefreshJobs 'Refresh Job Map
151 End Sub
152 Sub Map_SelectOrder()
153 With JobMap
154     OrderID = Replace(Application.Caller, Left(Application.Caller, 7), "" )
155     .Range("BM4" ).Value = OrderID
156     On Error Resume Next
157     .Shapes("SelJobGrp" & OrderID).Ungroup 'Ungroup if necessary
158     .Shapes("EditJobGrp" & OrderID).Delete 'Delete if it exists
159     On Error GoTo 0
160     .Shapes("JobEditSample" ).Duplicate.Name = "EditJobGrp" & OrderID
161
162     With .Shapes("EditJobGrp" & OrderID)
163         .GroupItems("SaveJobSample" ).Name = "SaveJob" & OrderID
164         .GroupItems("EditJobSample" ).Name = "EditJob" & OrderID
165         .GroupItems("CloseJobSample" ).Name = "CloseJob" & OrderID
    
```

```

166     .Left = JobMap.Shapes("JobGrp" & OrderID).Left + JobMap.Shapes("JobGrp" &
167         OrderID).Width - 40
168     .Top = JobMap.Shapes("JobGrp" & OrderID).Top
169     End With
170     .Shapes.Range(Array("JobGrp" & OrderID, "EditJobGrp" & OrderID)).Group.Name =
171         "SelJobGrp" & OrderID
172     .Shapes("SelJobGrp" & OrderID).Select
173     End With
174 End Sub
175
176 Sub Map_ShowRoute()
177     Dim Delay As Double, StartTime As Double
178     With OrdersDB
179         LastRow = .Range("AA9999" ).End(xlUp).Row
180         If LastRow < 3 Then Exit Sub
181         .Range("AA2:AH" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
182             "AJ2:AL3" ), CopyToRange:=.Range("AN2:AS2" ), Unique:=True
183         LastResultRow = .Range("AN9999" ).End(xlUp).Row
184         If LastResultRow < 3 Then Exit Sub
185         If LastResultRow < 4 Then GoTo NoSort
186         'On Error Resume Next
187         With .Sort 'Sort based on earliest time first
188             .SortFields.Clear
189             .SortFields.Add Key:=OrdersDB.Range("AP3" ), SortOn:=xlSortOnValues, Order:=
190                 xlAscending, DataOption:=xlSortNormal
191             .SetRange OrdersDB.Range("AN3:AS" & LastResultRow)
192             .Apply
193         End With
194         'On Error GoTo 0
195         NoSort:
196         TechNm = .Range("AJ3" ).Value 'Technician Name
197         JobMap.Shapes("CarRouteSample" ).Duplicate.Name = "TechRoute" & TechNm
198         For ResultRow = 3 To LastResultRow - 1
199             OrderID = .Range("AN" & ResultRow).Value 'Order ID
200             NextOrderID = .Range("AN" & ResultRow + 1).Value 'Next Order ID
201             StartLeft = .Range("AQ" & ResultRow).Value + 50 'Starting Left Position
202             StartTop = .Range("AR" & ResultRow).Value 'Starting Top Position
203             NextLeft = .Range("AQ" & ResultRow + 1).Value + 50 'Next Left Position
204             NextTop = .Range("AR" & ResultRow + 1).Value 'Next Top Position
205             LeftPos = StartLeft 'Set Intial Left Position
206             TopPos = StartTop 'Set Intial Top Position
207             With JobMap
208                 'Add in Connection
209                 .Shapes("ConSample" ).Duplicate.Name = "JobConn" & OrderID & "_" &
210                     NextOrderID
211                 With .Shapes("JobConn" & OrderID & "_" & NextOrderID).ConnectorFormat
212                     .BeginConnect JobMap.Shapes("MapIcon" & OrderID), 3
213                     .EndConnect JobMap.Shapes("MapIcon" & NextOrderID), 3
214                 End With
215                 'Drive the car
216                 For MoveCar = 1 To 50
217                     With .Shapes("TechRoute" & TechNm)
218                         .Left = LeftPos 'Set Initial Left
219                         .Top = TopPos 'Set Intial Top
220                     End With
221                     Delay = 0.006
222                     StartTime = Timer
223                     Do
224                         DoEvents
225                     Loop While Timer - StartTime < Delay
226                     LeftPos = StartLeft + (((NextLeft - StartLeft) / 50) * MoveCar)

```

```
1 2 3 4 5
222 'Increment Left Position
    TopPos = StartTop + (((NextTop - StartTop) / 50) * MoveCar) 'Increment
    Top Position
223     Next MoveCar
224     End With
225     Next ResultRow
226     JobMap.Shapes("TechRoute" & TechNm).Delete
227     'Delete connectors
228     For Each JobShp In JobMap.Shapes
229         If InStr(JobShp.Name, "JobConn" ) > 0 Then JobShp.Delete
230     Next JobShp
231     End With
232
233 End Sub
234 Sub Map_Today()
235     JobMap.Range("E2" ).Value = Date 'Current Day
236 End Sub
```



```

1 Option Explicit
2 Dim OrdRow As Long, OrdCol As Long, OrdItemDBRow As Long, LastRow As Long, ItemRow As Long, OrderRow As Long
3 Dim EquipFolder As String, FilePath As String, SelRow As Long, LastResultRow As Long, ResultRow As Long, AttRow As Long, AttDBRow As Long
4 Sub Cust_SavedMsg()
5     With Orders.Shapes("CustSavedMsg" )
6         Dim i As Long, Delay As Double, StartTime As Double
7         .Visible = msoCTrue
8         For i = 1 To 150
9             .Fill.Transparency = i / 150
10            Delay = 0.009
11            StartTime = Timer
12            Do
13                DoEvents
14            Loop While Timer - StartTime < Delay
15        Next i
16        .Visible = msoFalse
17    End With
18 End Sub
19
20
21 Sub Customer_SaveUpdate()
22     Dim CustRow As Long
23     With Orders
24         If .Range("B9" ).Value = Empty Then 'New Customer
25             CustRow = Customers.Range("B99999" ).End(xlUp).Row + 1 'First Avail Row
26             Customers.Range("B" & CustRow).Value = .Range("B10" ).Value 'Next Cust ID
27         Else 'Existing Customer
28             CustRow = .Range("B9" ).Value 'Existing Customer Row
29         End If
30         Customers.Range("C" & CustRow).Value = .Range("H6" ).Value 'Customer Name
31         Customers.Range("D" & CustRow).Value = .Range("H7" ).Value 'Address
32         Customers.Range("E" & CustRow).Value = .Range("H8" ).Value 'City
33         Customers.Range("F" & CustRow).Value = .Range("J8" ).Value 'State
34         Customers.Range("G" & CustRow).Value = .Range("L8" ).Value 'Zip
35         Customers.Range("H" & CustRow).Value = .Range("L9" ).Value 'Phone
36         Customers.Range("I" & CustRow).Value = .Range("H9" ).Value 'Email
37         .Range("H4" ).Value = .Range("H6" ).Value 'Update Cusotmer name
38     End With
39     Cust_SavedMsg
40 End Sub
41
42 Sub Order_Delete()
43     If MsgBox("Are you sure you want to delete this order?" , vbYesNo, "Delete Appointment" ) = vbNo Then Exit Sub
44     With Orders
45         If .Range("B4" ).Value = Empty Then GoTo NotSaved
46         OrdRow = .Range("B4" ).Value 'Order Row
47         OrdersDB.Range(OrdRow & ":" & OrdRow).EntireRow.Delete
48         'Clear Item
49         With OrderItemDB
50             LastRow = .Range("A99999" ).End(xlUp).Row
51             If LastRow < 3 Then GoTo NotSaved
52             .Range("A3:H" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("K2:K3" ), CopyToRange:=.Range("M2:S2" ), Unique:=True
53             LastResultRow = .Range("M99999" ).End(xlUp).Row
54             If LastResultRow < 3 Then GoTo NotSaved
55             If LastResultRow < 4 Then GoTo SingleRow
56             'Sort based on descending rows
57             With .Sort

```

1 2 3 4

```

58     1 2 3 4     .SortFields.Clear
59     .SortFields.Add Key:=OrderItemDB.Range("S3" ), SortOn:=xlSortOnValues,
        .Range OrderItemDB.Range("M3:S" & LastResultRow)
60     .SetRange OrderItemDB.Range("M3:S" & LastResultRow)
61     .Apply
62     End With
63     SingleRow:
64     For ResultRow = 3 To LastResultRow
65         OrdItemDBRow = .Range("S" & ResultRow).Value 'Equip Order DB Row
66         .Range(OrdItemDBRow & ":" & OrdItemDBRow).EntireRow.Delete 'Delete Order
        Item Row
67     Next ResultRow
68     End With
69     NotSaved:
70     Order_New 'Clear Out All Form Fields
71     Orders_LoadList 'Reload order list
72     End With
73 End Sub

74
75 Sub Order_Load()
76     With Orders
77         .Range(
78             "H4,H5,K5:L6,H7:L7,H8,J8,L8,H9:J9,L9,H11,J11,L11,H12,J12,L12,H13:L14,B18:B99,G18:L
79             999,B12,B13" ).ClearContents
80         If .Range("B4" ).Value = Empty Then
81             MsgBox "Please make sure to select a correct order"
82             Exit Sub
83         End If
84         Application.ScreenUpdating = False
85         OrdRow = .Range("B4" ).Value 'Order Row
86         For OrdCol = 2 To 10
87             .Range(OrdersDB.Cells(1, OrdCol).Value).Value = OrdersDB.Cells(OrdRow, OrdCol)
88             .Value 'Add/Update Data
89         Next OrdCol
90         .Range("H4" ).Value = .Range("H6" ).Value 'Trigger Customer Address Load
91
92         'Load Item
93         With OrderItemDB
94             LastRow = .Range("A99999" ).End(xlUp).Row
95             If LastRow < 4 Then GoTo NoData
96             .Range("A3:H" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
97                 "K2:K3" ), CopyToRange:=.Range("M2:S2" ), Unique:=True
98             LastResultRow = .Range("M99999" ).End(xlUp).Row
99             If LastResultRow < 3 Then GoTo NoData
100            For ResultRow = 3 To LastResultRow
101                OrderRow = .Range("M" & ResultRow).Value 'Set Order Row
102                Orders.Range("B" & OrderRow).Value = .Range("S" & ResultRow).Value 'Load
103                DB Row
104                Orders.Range("G" & OrderRow & ":K" & OrderRow).Value = .Range("N" &
105                    ResultRow & ":R" & ResultRow).Value 'Add Item, Desc.,Qty & Comp.
106            Next ResultRow
107            NoData:
108        End With
109    End With
110    Application.ScreenUpdating = True
111 End Sub

112
113 Sub Order_New()
114     With Orders
115         .Range(
116             "B3,B7,H4,H5,K6:L6,H6,H7:L7,H8,J8,L8,H9:J9,L9,H11,J11,L11,H12,J12,L12,H13:L14,B18:
117             "

```

```

1 2
110     B99,G18:L999,B12,B13" ).ClearContents
111     .Range("H4" ).Select
112 End With
112 End Sub
113 Sub Order_Print()
114     Orders.PrintOut , , , , True, , , , False
115 End Sub
116 Sub Order_SavedMsg()
117     With Orders.Shapes("OrdSavedMsg" )
118         Dim i As Long, Delay As Double, StartTime As Double
119         .Visible = msoCTrue
120         For i = 1 To 150
121             .Fill.Transparency = i / 150
122             Delay = 0.009
123             StartTime = Timer
124             Do
125                 DoEvents
126                 Loop While Timer - StartTime < Delay
127             Next i
128             .Visible = msoFalse
129         End With
130     End Sub
131 Sub Order_SaveUpdate()
132     With Orders
133         If .Range("H6" ).Value = Empty Then
134             MsgBox "Please make sure to add a Customer before saving this order"
135             Exit Sub
136         End If
137         If .Range("K6" ).Value = Empty Then
138             MsgBox "Please make sure to add a Job Name before saving this order"
139             Exit Sub
140         End If
141         If .Range("B4" ).Value = Empty Then 'New Order
142             OrdRow = OrdersDB.Range("A99999" ).End(xlUp).Row + 1 'New Database Row
143             .Range("B3" ).Value = .Range("B5" ).Value 'Set Next Order ID
144             OrdersDB.Range("A" & OrdRow).Value = .Range("B3" ).Value 'Set Order ID
145             OrdersDB.Range("K" & OrdRow).Value = .Range("B12" ).Value 'Left Location (if
146             any)
147             OrdersDB.Range("L" & OrdRow).Value = .Range("B13" ).Value 'Top Location (if
148             any)
149         Else 'Existing Order
150             OrdRow = .Range("B4" ).Value 'Existing Order Row
151         End If
152         For OrdCol = 2 To 10
153             OrdersDB.Cells(OrdRow, OrdCol).Value = .Range(OrdersDB.Cells(1, OrdCol).Value)
154             .Value 'Add/Update Data
155         Next OrdCol
156     End With
157     'Save Item
158     LastRow = .Range("H99" ).End(xlUp).Row
159     If LastRow < 18 Then GoTo NoItems
160     For ItemRow = 18 To LastRow
161         If .Range("B" & ItemRow).Value = Empty Then 'New DB Row
162             OrdItemDBRow = OrderItemDB.Range("A99999" ).End(xlUp).Row + 1 'First
163             avail Row
164             .Range("B" & ItemRow).Value = OrdItemDBRow 'Set DB Row
165             OrderItemDB.Range("A" & OrdItemDBRow).Value = .Range("B3" ).Value 'Set
166             Order ID
167             OrderItemDB.Range("F" & OrdItemDBRow).Value = ItemRow 'Set Order Item Row
168             OrderItemDB.Range("G" & OrdItemDBRow).Value = "=Row()" 'Set Formula Row

```

```

164     1 2 3 4
165     }Else 'Existing DB Row
166         OrdItemDBRow = .Range("B" & ItemRow).Value 'Order DB Row
167     End If
168     OrderItemDB.Range("B" & OrdItemDBRow).Value = .Range("G" & ItemRow).Value
169     'Item Name
170     OrderItemDB.Range("C" & OrdItemDBRow).Value = .Range("H" & ItemRow).Value
171     'Description
172     OrderItemDB.Range("D" & OrdItemDBRow).Value = .Range("J" & ItemRow).Value
173     'Qty
174     OrderItemDB.Range("E" & OrdItemDBRow).Value = .Range("K" & ItemRow).Value
175     'Completed On/At
176     Next ItemRow
177     NoItems:
178     Order_SavedMsg 'Run Fade Out Message
179     DoEvents
180 End With
181 End Sub
182
183 Sub Orders_ClearFilter()
184     Orders.Range("E5" ).Value = "Enter Search"
185 End Sub
186
187 Sub Orders_LoadList()
188     Orders.Range("B7,D7:E9999" ).ClearContents
189     With OrdersDB
190         LastRow = .Range("A99999" ).End(xlUp).Row
191         If LastRow < 4 Then Exit Sub
192         .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
193             "O2:Q3" ), CopyToRange:=.Range("S2:T2" ), Unique:=True
194         LastResultRow = .Range("S99999" ).End(xlUp).Row
195         If LastResultRow < 3 Then Exit Sub
196         Orders.Range("D7:E" & LastResultRow + 4).Value = .Range("S3:T" & LastResultRow)
197         .Value 'Bring in Order ID and Customer Name
198     End With
199 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
           CalPopUp Sheet
32 End Sub
33
34 Sub CalendarShow()
35     With ActiveSheet
36         Set SelCell = Selection
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
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" ).
           Value) 'Set Month
56         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
           Value) 'Set Year
57         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
58         ' UnGroupCal

```

1 2

```

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

```

```

1
113 CreateWS:
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

```

1 2

```

1 2
170 .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) & ")"
171 'Set Defined Names
172 ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
173 ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
174
175 'Add in Calendar Formulas
176
177 .Range("B1" ).Value =
    "=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,Ca
lMonths,0),1)," & Chr(34) & Chr(34) & ")"
178 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
    ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
179 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
    ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
180 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
    ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
181 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
    ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
182 .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
    ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
183 .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
    ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
184 .Range("B2" ).Value = "=H1+1"
185 .Range("C2" ).Value = "=B2+1"
186 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
187 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
188 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
    & Chr(34) & Chr(34) & ",H4+1)"
189 .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",B5+1)," & Chr(34) & Chr(34) & ")"
190 .Range("B6" ).Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) &
    ",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) &
    ")"
191 .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",I5+1)," & Chr(34) & Chr(34) & ")"
192 .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
193
194 'Set format to Single Day
195 .Range("B1:H6" ).NumberFormat = "d"
196
197 'Add in relative Day #'s
198 .Range("B7" ).Value = "1"
199 .Range("C7" ).Value = "2"
200 .Range("B8" ).Value = "8"
201 .Range("C8" ).Value = "9"
202 .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
203 .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
204 GroupCal
205 End With
206 Exit Sub
207 NoCal:
208 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
    calendar over from another sheet and paste into this sheet"

```



```

1
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
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
255
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

```

1 2 3

```

1 2 3
313     .Range("A3" ).Value = 12
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" ,
320     "40Day" , "41Day" , "42Day" )).Visible = True
321     Else:
322     ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
323     "40Day" , "41Day" , "42Day" )).Visible = False
324     End If
325     End With
326     End Sub
327
328 Sub PrevYear()
329     ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
330     "CalPopUp" ).Range("A2" ).Value - 1
331 End Sub
332
333 Sub ReplaceCalendar() 'Shape Deleted
334     CalPopUp.Shapes("Calendar" ).Copy 'Copy From Developers sheet
335     ActiveCell.Select 'Select the active Cell
336     ActiveSheet.Paste 'Paste in Calendar
337 End Sub
338
339 .....
340
341 ''''Select Day Of The Month
342 .....
343
344 Sub SelectDay()
345     Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
346     DayNumb = Replace(Application.Caller, "Day" , "" )
347     RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
348     ColNumb = DayNumb Mod 7 + 1
349     If DayNumb Mod 7 = 0 Then
350     ColNumb = 8
351     Else
352     ColNumb = DayNumb Mod 7 + 1
353     End If
354
355     'On Error Resume Next
356     If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
357     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
358     ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
359     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
360     ActiveCell.Offset(0, 1).Select
361 End Sub
362
363 Sub ShowSettings()
364     'Show or Hide Calendar Settings Panel
365     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
366     ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
367     "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
368     = False
369     Else:
370     ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
371     "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
372     = True
373     End If
374 End Sub
375
376 Sub UnGroupCal()
377     On Error Resume Next

```

```
1
366 ActiveSheet.Shapes("Calendar").Ungroup
367 ActiveSheet.Shapes("NextMonth").Ungroup
368 ActiveSheet.Shapes("PrevMonth").Ungroup
369 On Error GoTo 0
370 End Sub
```

- ..., 30
- A**
- Activate, 16, 17, 28
 - ActiveCell, 16, 27, 32
 - ActiveSheet, 26-28, 30-33
 - ActiveWorkbook, 27-29
 - ActSht, 28
 - Add, 20, 23, 28, 29
 - Address, 27, 28
 - Admin, 16, 18
 - Admin_SetStatusColor, 16
 - AdvancedFilter, 17, 20, 22, 23, 25
 - After, 28
 - AllowMultiSelect, 16
 - Application, 16, 17, 19, 23, 26, 32
 - Apply, 20, 23
 - Array, 19, 20, 26, 27, 30-32
 - AttDBRow, 22
 - AttRow, 22
 - AutoFill, 28, 29
 - AutoSize, 18
- B**
- BeginConnect, 20
 - Bold, 26, 27, 31
 - BrowseForMapFolder, 16
 - BrowseForTechFolder, 16
- C**
- Cal, 9, 12
 - CalCol, 26, 28
 - CalendarHide, 9, 12, 26
 - CalendarShow, 9, 12, 26
 - CalFormulaReplacement, 27
 - Caller, 16, 17, 19, 26, 32
 - CalPopUp, 32
 - Cells, 23, 24, 27, 28, 32
 - CheckForSheet, 9, 12, 27
 - Chr, 3, 28, 29
 - Clear, 20, 23
 - ClearContents, 3, 11, 17, 23-25
 - ColCnt, 28
 - ColNum, 27
 - ColNumb, 32
 - Color, 16, 18
 - ConnectorFormat, 20
 - Copy, 28, 32
 - CopyToRange, 17, 20, 22, 23, 25
 - Count, 28
 - CountLarge, 3, 11, 12
 - CreateCalSht, 28
 - CreateWS, 27, 28
 - CriteriaRange, 17, 20, 22, 23, 25
 - Cust_SavedMsg, 22
 - Customer_SaveUpdate, 22
 - Customers, 11, 19, 22
 - CustRow, 11, 19, 22
- D**
- DataOption, 20, 23
 - DayCnt, 28
 - DayName, 26, 27, 31
 - DayNum, 26, 27
 - DayNumb, 32
 - Delay, 20, 22, 24
 - Delete, 17, 19, 21-23
 - Destination, 28, 29
 - Dir, 18
 - DoEvents, 20, 22, 24, 25
 - DrawingObject, 27, 28
 - Duplicate, 18-20
- E**
- Empty, 3, 11, 12, 19, 22-24, 26, 31, 32
 - EndConnect, 20
 - EntireRow, 22, 23
 - EquipFolder, 22
 - Explicit, 3-11, 13-17, 22
- F**
- FileDialog, 16
 - FilePath, 22
 - Fill, 16, 18, 22, 24, 26, 27, 31
 - Find, 11, 18
 - Font, 26, 27, 31
 - ForeColor, 16, 18, 26, 27, 31
 - Formula, 17, 27, 28
- G**
- Group, 19, 20, 30
 - GroupCal, 29, 30
 - GroupItems, 19, 27
- H**
- Height, 18
 - Home, 16
- I**
- i, 22, 24
 - InStr, 17, 21, 27, 31
 - Interior, 16, 18
 - Intersect, 3, 9, 11, 12
 - IsDate, 26
 - ItemDBRow, 11
 - ItemRow, 22, 24, 25
 - Items, 11
- J**
- JobFolder, 16
 - JobMap, 17-21
 - JobNm, 17, 18
 - JobShp, 17, 19, 21
- K**
- Key, 20, 23
- L**
- LastResultRow, 17, 18, 20, 22, 23, 25
 - LastRow, 17, 20, 22-25
 - Left, 3, 18-20, 27, 31
 - LeftPos, 17, 18, 20
 - Len, 31
 - Line, 18
- M**
- MacroLink, 30, 31
 - MacroLinkRemover, 27, 30
 - Map_CloseEdit, 17
 - Map_EditJob, 17
 - Map_NextDay, 17
 - Map_PrevDay, 17
 - Map_RefreshJobs, 9, 17, 19
 - Map_SaveAsCustLoc, 19
 - Map_SaveLocation, 19
 - Map_SelectOrder, 19
 - Map_ShowRoute, 20
 - Map_Today, 21
 - MapFolder, 17, 18
 - MapPath, 17, 18
 - MapPic, 17, 18
 - Menu_GoHome, 16
 - Menu_GoToSheet, 16
 - Month, 26, 28
 - MoveCar, 17, 20, 21
 - MsgBox, 11, 19, 22-24, 27, 29
 - msoAutoSizeShapeToFitText, 18
 - msoBringToFront, 27
 - msoCTrue, 3, 22, 24, 27, 30
 - msoFalse, 3, 16, 22, 24, 26, 31, 32
 - msoFileDialogFolderPicker, 16
 - msoTrue, 27
- N**
- Name, 17-21, 28-30
 - Names, 29
 - NewLink, 30, 31
 - NextLeft, 17, 20
 - NextMonth, 31
 - NextOrderID, 17, 20
 - NextShp, 30, 31
 - NextTop, 17, 20, 21
 - NextYear, 31
 - NoCal, 26, 27, 29
 - NoData, 23
 - NoItems, 24, 25
 - NoResults, 17
 - NoSelection, 16
 - NoSort, 20
 - NotSaved, 22, 23
 - NumberFormat, 29
- O**
- Offset, 27, 32
 - OnAction, 27, 28, 30, 31
 - OrdCol, 22-24
 - Order, 20, 23
 - Order_Delete, 22
 - Order_Load, 12, 17, 23
 - Order_New, 23
 - Order_Print, 24
 - Order_SavedMsg, 24, 25
 - Order_SaveUpdate, 24
 - OrderID, 17-20
 - OrderItemDB, 22-25
 - OrderRow, 17, 19, 22, 23
 - Orders, 17, 22-25
 - Orders_ClearFilter, 25
 - Orders_LoadList, 11, 23, 25
 - OrdersDB, 17-20, 22-25
 - OrdItemDBRow, 22-25
 - OrdRow, 22-24
- P**
- Paste, 28, 32
 - PicFolder, 17, 18
 - PicPath, 17, 18
 - Placement, 27, 30
 - PrevMonth, 31
 - PrevYear, 32
 - PrintOut, 24
- R**
- Range, 3, 9, 11, 12, 16-32
 - RefersTo, 29
 - Replace, 17, 19, 32
 - ReplaceCalendar, 9, 12, 32
 - ResultRow, 17-23

RGB, [16](#), [18](#), [26](#), [27](#), [31](#)
Right, [31](#)
RoundUp, [32](#)
Row, [3](#), [11](#), [12](#), [17](#), [18](#), [20](#), [22-25](#)
RowCnt, [28](#)
RowNum, [27](#)
RowNumb, [32](#)

S

ScreenUpdating, [16](#), [23](#)
SelCell, [26](#), [27](#)
SelectDay, [32](#)
SelectedItems, [16](#)
Selection, [26](#), [30](#)
SelRow, [22](#)
SetBackgroundPicture, [18](#)
SetRange, [20](#), [23](#)
Shape, [9](#), [12](#), [17](#), [27](#), [30](#)
ShapeRange, [26](#), [30](#)
Shapes, [3](#), [9](#), [12](#), [16-22](#), [24](#), [26-28](#), [30-33](#)
Sheets, [16](#), [26-28](#), [31](#), [32](#)
Show, [16](#)
ShowSettings, [32](#)
Shp, [27](#), [30](#), [31](#)
SingleRow, [22](#), [23](#)
Sort, [20](#), [22](#)
SortFields, [20](#), [23](#)
SortOn, [20](#), [23](#)
Split, [31](#)
SplitLink, [30](#), [31](#)
StartLeft, [17](#), [20](#)
StartTime, [20](#), [22](#), [24](#)
StartTop, [17](#), [20](#), [21](#)
Status, [17](#), [18](#)
StatusColor, [17-19](#)
StatusRow, [17](#), [18](#)

T

Target, [3](#), [9](#), [11](#), [12](#)
TecFolder, [17](#)
TechNm, [17](#), [20](#), [21](#)
Text, [18](#)
TextFrame2, [18](#), [26](#), [27](#), [31](#)
TextRange, [18](#), [26](#), [27](#), [31](#)
ThisWorkbook, [28](#), [31](#), [32](#)
Timer, [20](#), [22](#), [24](#)
Title, [16](#)
Top, [3](#), [18-20](#), [27](#)
TopPos, [17](#), [18](#), [20](#), [21](#)
Transparency, [22](#), [24](#)

U

Ungroup, [17](#), [19](#), [33](#)
UnGroupCal, [28](#), [32](#)
Unique, [17](#), [20](#), [22](#), [23](#), [25](#)
UnMapTop, [17](#), [18](#)
UserPicture, [18](#)

V

Value, [3](#), [11](#), [12](#), [16-29](#), [31](#), [32](#)
vbDirectory, [18](#)
vbNo, [22](#)
vbYesNo, [22](#)
Visible, [3](#), [9](#), [12](#), [16](#), [22](#), [24](#), [26-28](#), [30-32](#)

W

Width, [20](#)
Worksheet, [27](#), [28](#)
Worksheet_Change, [9](#), [11](#)
Worksheet_SelectionChange, [3](#), [9](#), [11](#)

WorksheetFunction, [32](#)
ws, [27](#), [28](#)

X

xlAscending, [20](#)
xlDescending, [23](#)
xlFillDefault, [28](#), [29](#)
xlFilterCopy, [17](#), [20](#), [22](#), [23](#), [25](#)
xlMove, [27](#), [30](#)
xlSheetHidden, [28](#)
xlSortNormal, [20](#), [23](#)
xlSortOnValues, [20](#), [23](#)
xlUp, [17](#), [20](#), [22-25](#)
xlValues, [11](#), [18](#)
xlWhole, [11](#), [18](#)

Y

Year, [26](#), [28](#)

Z

ZOrder, [27](#)

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