

# VBA SOURCE CODE BOOK

## VERTICAL SCHEDULING CALENDAR

**UPDATED**

**EXCLUSIVE FOR SILVER & GOLD MEMBERS**

**FEATURE, FIX OR FOCUS**



*Excel For Freelancers*

## Create This Vertical Scheduling Calendar In Excel



**DOWNLOAD  
APPLICATION**



**VIEW  
TRAINING**

*by: Randy Austin*

# ABOUT THE AUTHOR

A 4-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With over 471,000 YouTube subscribers, 35,448,742 video views, 430+ comprehensive training videos, and a thriving 65,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



Microsoft®  
Most Valuable  
Professional



# OUR COURSES & PRODUCTS



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

[Click here to learn more](#)



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

[Click here to learn more](#)

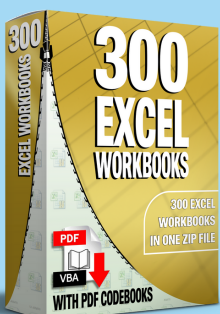
-



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

[Click here to learn more](#)

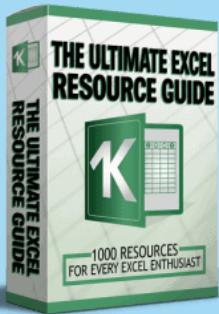
-



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

[Click here to learn more](#)

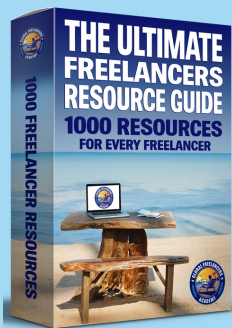
# OUR COURSES & PRODUCTS



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

[Click here to learn more](#)

-



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

[Click here to learn more](#)

-



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

[Click here to learn more](#)

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Worksheet_Change [Sub ].....	2
Worksheet_SelectionChange [Sub ].....	2
ApptsDB.....	3
(Declarations).....	3
Calendar.....	4
(Declarations).....	4
Worksheet_SelectionChange [Sub ].....	4
Sheet1.....	5
(Declarations).....	5
ThisWorkbook.....	6
(Declarations).....	6
Modules.....	7
Admin_Macros.....	7
(Declarations).....	7
Admin_SetTypeColor [Sub ].....	7
Appt_Macros.....	8
(Declarations).....	8
Appt_Delete [Sub ].....	8
Appt_Load [Sub ].....	8
Appt_New [Sub ].....	8
Appt_SaveUpdate [Sub ].....	8
Calendar_Macros.....	10
(Declarations).....	10
Calendar_Appt_Select [Sub ].....	10
Calendar_ChangeView [Sub ].....	10
Calendar_GoToMonth [Sub ].....	10
Calendar_GoToWeek [Sub ].....	10
Calendar_NextMonth [Sub ].....	11
Calendar_NextWeek [Sub ].....	11
Calendar_PrevMonth [Sub ].....	11
Calendar_PrevWeek [Sub ].....	11
Calendar_Refresh [Sub ].....	11
Calendar_ThisMonth [Sub ].....	12
Calendar_ThisWeek [Sub ].....	13
DragDrop_Macros.....	14
(Declarations).....	14
Appt_CheckForMove [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

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F8" )) Is Nothing And Range("F8" ).Value <> Empty
5         Then
6             Calendar.Range("D6:D371" ).NumberFormat = Target.Value 'Change Date Format
7         End If
8         If Not Intersect(Target, Range("F9" )) Is Nothing And Range("F9" ).Value <> Empty
9             Then Calendar_ChangeView 'View Change
10        End Sub
11
12 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
13     If Shapes("ColorPalette" ).Visible = True Then Shapes("ColorPalette" ).Visible =
14         msoFalse
15
16     'On Appt. Type Color Selection
17     If Not Intersect(Target, Range("C15:C33" )) Is Nothing Then
18         With Shapes("ColorPalette" )
19             .Left = Range("C" & Target.Row).Left 'Set Left Position
20             .Top = Range("C" & Target.Row + 1).Top 'Top Position
21             .Visible = msoCTrue
22         End With
23     End If
24
25     'On workday selection
26     If Not Intersect(Target, Range("C5:C11" )) Is Nothing Then
27         If Target.Value = Chr(111) Then Target.Value = Chr(254) Else Target.Value = Chr(
28             111)
29         Range("F4" ).Select
30     End If
31 End Sub
```

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
      it in
5     Dim Cal As Shape
6     On Error Resume Next
7     Set Cal = Shapes("Calendar" )
8     On Error GoTo 0
9     If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
10    If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
11
12    If Not Intersect(Target, Range("N3" )) Is Nothing Then 'Change this to any cell (or
      cells) you would like to have the Pop-Up Calendar Appear
13        CheckForSheet
14        CalendarShow
15    Else:
16        CheckForSheet
17        CalendarHide
18    End If
19 End Sub
```



1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2
3 Sub Admin_SetTypeColor()
4     ActiveCell.Interior.Color = Admin.Shapes(Application.Caller).Fill.ForeColor.RGB
5     Admin.Shapes("ColorPalette").Visible = msoFalse
6 End Sub
```

```

1 Option Explicit
2
3 Dim ApptRow As Long, ApptCol As Long
4
5 Sub Appt_Delete()
6     If MsgBox("Are you sure you want to delete this appointment?" , vbYesNo, "Delete
7         Appointment" ) = vbNo Then Exit Sub
8     With Calendar
9         If .Range("B3" ).Value = Empty Then GoTo NotSaved
10        ApptRow = .Range("B3" ).Value 'Appt DB Row
11        ApptsDB.Range(ApptRow & ":" & ApptRow).EntireRow.Delete
12        NotSaved:
13        Appt_New
14        Calendar_Refresh 'Run macro to refresh calendar
15    End With
16 End Sub
17
18 Sub Appt_Load()
19     With Calendar
20         If .Range("B3" ).Value = Empty Then
21             MsgBox "Please select a correct appointment to load"
22             Exit Sub
23         End If
24         Application.ScreenUpdating = False
25         ApptRow = .Range("B3" ).Value 'Appt DB Row
26         For ApptCol = 2 To 7
27             .Range(ApptsDB.Cells(1, ApptCol).Value).Value = ApptsDB.Cells(ApptRow, ApptCol
28                 ).Value 'Bring Over Data
29         Next ApptCol
30         Application.ScreenUpdating = True
31     End With
32 End Sub
33
34 Sub Appt_New()
35     Calendar.Range("B2,G3:J3,N3:O3,R3:S3,W3,AA3:AC3,AG3:A03" ).ClearContents 'Clear Out
36     Existing Cells
37 End Sub
38
39 Sub Appt_SaveUpdate()
40     With Calendar
41         If Application.WorksheetFunction.CountA(.Range("G3,N3,R3,W3" )) < 4 Then
42             MsgBox "Please fill in the required fields, Item, Date, Time & Duration"
43             Exit Sub
44         End If
45         Application.ScreenUpdating = False
46         If .Range("B3" ).Value = Empty Then 'New Appt
47             ApptRow = ApptsDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail. Row
48             .Range("B2" ).Value = .Range("B4" ).Value 'Next Appt ID
49             ApptsDB.Range("A" & ApptRow).Value = .Range("B2" ).Value 'Appt ID
50         Else 'Existing
51             ApptRow = .Range("B3" ).Value
52         End If
53         For ApptCol = 2 To 7
54             ApptsDB.Cells(ApptRow, ApptCol).Value = .Range(ApptsDB.Cells(1, ApptCol).Value
55                 ).Value 'Bring Over Data
56         Next ApptCol
57         Calendar_Refresh 'Run macro to refresh calendar
58         Application.ScreenUpdating = True
59     End With
60 End Sub

```

```
57 | 1 2 End With  
58 | End Sub
```

```

1 Option Explicit
2
3 Dim ApptRow As Long, ApptCol As Long, LastRow As Long, LastResultRow As Long
4 Dim DurCols As Long, ResultRow As Long, TypeRow As Long
5 Dim ApptColor As String, ApptID As String, ApptName As String, ApptType As String
6 Dim ApptShp As Shape
7 Dim ApptDate As Date, StartDate As Date
8 Dim ApptLeft As Double, ApptTime As Double, ApptDur As Double, SchedDur As Double,
  SchedStart As Double
9
10
11 Sub Calendar_Appt_Select()
12     With Calendar
13         ApptID = Replace(Application.Caller, "CalAppt" , "" ) 'Extract Appt ID
14         .Range("B2" ).Value = ApptID
15         .Range("B13" ).Value = .Shapes(Application.Caller).Left 'Set Left Pos. of
  Current Selected Shape
16         .Range("B14" ).Value = .Shapes(Application.Caller).Top 'Set Top Pos of Sel.
  Shape
17         .Range("B15" ).Value = False 'Set Move Shape To False
18         .Shapes(Application.Caller).Select 'Select On shape
19         Appt_Load
20         Appt_CheckForMove
21     End With
22 End Sub
23
24 Sub Calendar_ChangeView()
25     With Calendar
26         If [ViewType] = "Month" Then
27             .Rows("6:371" ).EntireRow.RowHeight = 15.5
28             .Shapes("MonthGrp" ).Visible = msoCTrue
29             .Shapes("WeekGrp" ).Visible = msoFalse
30         Else 'Week View
31             .Range("6:371" ).EntireRow.RowHeight = 68
32             .Shapes("MonthGrp" ).Visible = msoFalse
33             .Shapes("WeekGrp" ).Visible = msoCTrue
34         End If
35     End With
36 End Sub
37
38 Sub Calendar_GoToMonth()
39     Dim MonthStart As Date
40     Dim CalRow As Long
41     With Calendar
42         MonthStart = DateSerial(.Range("B5" ).Value, .Range("B6" ).Value, 1)
43         CalRow = DatePart("y" , MonthStart) + 5 'Calendar Row based on the day # of the
  year
44         If CalRow > 5 Then ActiveWindow.ScrollRow = CalRow
45     End With
46 End Sub
47
48 Sub Calendar_GoToWeek()
49     Dim WeekStart As Date
50     Dim CalRow As Long
51     With Calendar
52         WeekStart = .Range("B8" ).Value 'Week Start
53         CalRow = DatePart("y" , WeekStart) + 5 'Calendar Row based on the day # of the
  year
54         If CalRow > 5 Then ActiveWindow.ScrollRow = CalRow
55     End With

```

1

```

1
56 End Sub
57
58
59 Sub Calendar_NextMonth()
60 With Calendar
61 If .Range("B6" ).Value = 12 Then 'On December Go to Next year and January
62 .Range("B5" ).Value = .Range("B5" ).Value + 1 'Increase year by 1
63 .Range("B6" ).Value = 1 'Set To January
64 Calendar_Refresh 'Refresh Calendar
65 Else:
66 .Range("B6" ).Value = .Range("B6" ).Value + 1
67 End If
68 Calendar_GoToMonth 'Go to specific month
69 End With
70 End Sub
71
72 Sub Calendar_NextWeek()
73 With Calendar
74 If Year(.Range("B8" ).Value) <> Year(.Range("B8" ).Value + 7) Then 'On Next
year week, go to next year
75 .Range("B5" ).Value = .Range("B5" ).Value + 1 'Increase year by 1
76 .Range("B6" ).Value = 1 'Set To January
77 Calendar_Refresh 'Refresh Calendar
78 End If
79 .Range("B8" ).Value = .Range("B8" ).Value + 7
80 Calendar_GoToWeek 'Go to specific week
81 End With
82 End Sub
83
84 Sub Calendar_PrevMonth()
85 With Calendar
86 If .Range("B6" ).Value = 1 Then 'On January Go to Prev Year & December
87 .Range("B5" ).Value = .Range("B5" ).Value - 1 'Reduce year by 1
88 .Range("B6" ).Value = 12 'Set To December
89 Calendar_Refresh 'Refresh Calendar
90 Else:
91 .Range("B6" ).Value = .Range("B6" ).Value - 1
92 End If
93 Calendar_GoToMonth 'Go to specific month
94 End With
95 End Sub
96
97
98 Sub Calendar_PrevWeek()
99 With Calendar
100 If Year(.Range("B8" ).Value) <> Year(.Range("B8" ).Value - 7) Then 'On Previous
year week, go to previous year
101 .Range("B5" ).Value = .Range("B5" ).Value - 1 'Reduce year by 1
102 .Range("B6" ).Value = 12 'Set To December
103 Calendar_Refresh 'Refresh Calendar
104 End If
105 .Range("B8" ).Value = .Range("B8" ).Value - 7
106 Calendar_GoToWeek 'Go to specific week
107 End With
108 End Sub
109
110 Sub Calendar_Refresh()
111 'Clear All existing apt shapes
112 For Each ApptShp In Calendar.Shapes
113 If InStr(ApptShp.Name, "CalAppt" ) > 0 Then ApptShp.Delete
1 2

```

```

1 2
114 Next ApptShp
115
116 'Run advanced filter for appts. database
117 With ApptsDB
118     LastRow = .Range("A99999").End(xlUp).Row 'Last Row of data
119     If LastRow < 4 Then Exit Sub
120     .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
121         "I2:K3" ), CopyToRange:=.Range("M2:R2" ), Unique:=True
122     LastResultRow = .Range("M99999").End(xlUp).Row
123     If LastResultRow < 3 Then Exit Sub
124     Application.ScreenUpdating = False ' Turn Off Screen updating
125     SchedDur = [Duration] 'Schedule duration
126     SchedStart = [StartTime] 'Schedule Start Time
127     StartDate = [StartYearDate] 'Start Year Date
128     For ResultRow = 3 To LastResultRow
129         ApptID = .Range("M" & ResultRow).Value 'Appt ID
130         ApptName = .Range("N" & ResultRow).Value 'Appointment Name
131         ApptType = .Range("O" & ResultRow).Value 'Appt Type
132         ApptDate = .Range("P" & ResultRow).Value 'Appt Date
133         ApptTime = .Range("Q" & ResultRow).Value 'Appt Time
134         ApptDur = .Range("R" & ResultRow).Value 'Appt Duration
135         DurCols = Int(ApptDur / SchedDur) 'Set & of columns for duration
136         On Error Resume Next
137         TypeRow = Admin.Range("ApptTypes").Find(ApptType, , xlValues, xlWhole).Row
138         'Get Type Row
139         On Error GoTo 0
140         If TypeRow <> 0 Then ApptColor = Admin.Range("C" & TypeRow).Interior.Color
141         Else ApptColor = "" 'Get Interior Color
142         ApptRow = ApptDate - StartDate + 6
143         ApptCol = (ApptTime - SchedStart) / SchedDur + 5
144         If ApptCol > 53 Then GoTo NextAppt
145         Calendar.Shapes("SampleApptShp").Duplicate.Name = "CalAppt" & ApptID
146         With Calendar.Shapes("CalAppt" & ApptID)
147             .Left = Calendar.Cells(ApptRow, ApptCol).Left
148             .Top = Calendar.Cells(ApptRow, ApptCol).Top + 1
149             If ApptCol + DurCols > 53 Then DurCols = 53 - ApptCol
150             .Width = Range(Calendar.Cells(ApptRow, ApptCol), Calendar.Cells(ApptRow,
151                 ApptCol + DurCols)).Width 'Set Appt Width
152             .Height = Calendar.Cells(ApptRow, ApptCol).Height 'Set Height - 1
153             .TextFrame2.TextRange.Text = Format(ApptTime, "h:mm/p") & ": " &
154                 ApptName 'Text inside shape
155             If ApptColor <> "" Then .Fill.ForeColor.RGB = ApptColor 'Set Shape color
156             .OnAction = "Calendar_Appt_Select" 'Macro to run
157         End With
158     Next Appt:
159     Next ResultRow
160     Application.ScreenUpdating = True 'Turn on Screen updating
161 End With
162 End Sub
163
164 Sub Calendar_ThisMonth()
165 With Calendar
166     .Range("B6").Value = Month(Date) 'Set Current Month
167     If Year(Date) <> .Range("B5").Value Then
168         .Range("B5").Value = Year(Date) 'Set to current year
169         Calendar_Refresh 'Refresh Calendar
170     End If
171     Calendar_GoToMonth 'Go to specific month
172 End With
173 End Sub

```



```
169  
170 Sub Calendar_ThisWeek()  
171     Calendar.Range("B8").Value = .Range("B9").Value  
172     Calendar.GoToWeek 'Go to this week  
173 End Sub
```

```

1  Option Explicit
2
3  Sub Appt_CheckForMove()
4      Dim DestRow As Long, DestCol As Long, ApptRow As Long, ApptCol As Long, CountDelay
5      As Long
6      Dim ApptID As String, DestDate As String
7      Dim DestTime As Date
8      With Calendar
9          ApptID = .Range("B2").Value 'Select Appt ID
10         ApptRow = .Range("B3").Value 'Select Appt DB Row
11         For CountDelay = 1 To 100000
12             DoEvents
13             If .Range("B15").Value = True Then GoTo EndDragDrop 'Exit loop move Appt =
14                 True
15                 With .Shapes("CalAppt" & ApptID)
16                     If .Left <> Calendar.Range("B13").Value Or .Top <> Calendar.Range("B14").
17                         Value Then 'Shape has been moved
18                             'Check to see if shape is moved within proper grid area
19                             If .Left < Calendar.Range("E1").Left Or .Left > Calendar.Range("BB1").
20                                 Left Or .Top < Calendar.Range("E6").Top - 1 Or Calendar.Range("D" & .
21                                     TopLeftCell.Row).Value = Empty Then
22                                     MsgBox "Please move the Appt shape within the given Calendar"
23                                     GoTo EndDragDrop
24                                 End If
25                             End If
26                             DestDate = Calendar.Range("D" & .TopLeftCell.Row).Value 'Destination
27                             Date
28                             DestTime = Calendar.Cells(5, .TopLeftCell.Column).Value 'Destination
29                             Time
30                             ApptsDB.Range("D" & ApptRow).Value = DestDate 'Update Date
31                             ApptsDB.Range("E" & ApptRow).Value = DestTime 'Update Time
32                             GoTo EndDragDrop
33                         End If
34                     End With
35                 Next CountDelay
36             EndDragDrop:
37             Calendar_Refresh 'Refresh Calendar
38             Appt_Load 'Refresh Appt data
39             .Range("B15").Value = True
40         End With
41     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 = ActiveCell
37         'Check if active cell is a valid date
38         If IsDate(SelCell.Value) = True Then
39             Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
40         Else: 'If No Date or incorrect Date user current date
41             Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
42         End If
43         'Clear all shapes to white (if calendar is visible)
44         If ActiveSheet.Shapes("Calendar" ).Visible = True Then
45             For DayNum = 1 To 42
46                 DayName = DayNum & "Day"
47                 With ActiveSheet.Shapes(DayName)
48                     .Fill.ForeColor.RGB = RGB(255, 255, 255)
49                     .TextFrame2.TextRange.Font.Bold = msoFalse
50                 End With
51             Next DayNum
52         End If
53
54         Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).
           Value) 'Set Month
55         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
           Value) 'Set Year
56         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
57         ' UnGroupCal

```

1 2

```

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

```

```

1
112 Exit Sub
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)"

```

1 2

```

1 2
168 .Range("A8" ).Value = "January"
169 .Range("A8" ).AutoFill Destination:=.Range("A8:A19" ), Type:=xlFillDefault
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$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A
    $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
179 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
    ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A
    $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
180 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
    ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A
    $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
181 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
    ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A
    $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
182 .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
    ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A
    $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
183 .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
    ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A
    $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
184 .Range("B2" ).Value = "=H1+1"
185 .Range("C2" ).Value = "=B2+1"
186 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
187 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
188 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
    & Chr(34) & Chr(34) & ",H4+1)"
189 .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",B5+1)," & Chr(34) & Chr(34) & ")"
190 .Range("B6" ).Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) &
    ",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) &
    ")")
191 .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",I5+1)," & Chr(34) & Chr(34) & ")"
192 .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
193
194 'Set format to Single Day
195 .Range("B1:H6" ).NumberFormat = "d"
196
197 'Add in relative Day #'s
198 .Range("B7" ).Value = "1"
199 .Range("C7" ).Value = "2"
200 .Range("B8" ).Value = "8"
201 .Range("C8" ).Value = "9"
202 .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
203 .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
204 GroupCal
205 End With
206 Exit Sub

```

```

1
207 NoCal:
208 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
calendar over from another sheet and paste into this sheet"
209 End Sub
210 Sub GroupCal()
211 ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
212 Selection.ShapeRange.Name = "NextMonth"
213 ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
214 Selection.ShapeRange.Name = "PrevMonth"
215 ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
216 , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" ,
"1Day" , _
217 "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" ,
"6Day" , _
218 "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
"18Day" _
219 , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day"
, _
220 "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
"Sa" , _
221 "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
"CalCol3" , _
222 "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" ,
"PrevMonth" , _
223 "NextMonth" , "NextYr" , "PrevYr" ).Visible = msoCTrue
224 ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
225 , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day"
, _
226 "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
227 "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
"18Day" _
228 , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
229 "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
"Sa" , _
230 "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
"CalCol3" , _
231 "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth"
, _
232 "NextMonth" , "NextYr" , "PrevYr" )).Select
233 Selection.ShapeRange.Group.Select
234 Selection.ShapeRange.Name = "Calendar"
235 Selection.Name = "Calendar"
236 Selection.Placement = xlMove
237 ActiveSheet.Shapes("Calendar" ).Placement = 2
238 End Sub
239
240
241
242
243 Sub MacroLinkRemover()
244 'PURPOSE: Remove an external workbook reference from all shapes triggering macros
245 'Source: www.ExcelForFreelancers.com
246 Dim Shp As Shape
247 Dim MacroLink, NewLink As String
248 Dim SplitLink As Variant
249
250 For Each Shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
1 2

```

```

1 2
251
252 'Grab current macro link (if available)
253 On Error GoTo NextShp
254 MacroLink = Shp.OnAction
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 straggling 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         End With
311     End If
312 End Sub
1 2 3

```



```

1 2 3
308     .TextFrame2.TextRange.Font.Bold = msoFalse
309     End With
310 End If
311 With Sheets("CalPopUp" )
312     If .Range("A3" ).Value = 1 Then
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     ThisWorkbook.Sheets("CalPopUp" ).Shapes("Calendar" ).Copy 'Copy From Developers
335     sheet
336     ActiveCell.Select 'Select the active Cell
337     ActiveSheet.Paste 'Paste in Calendar
338 End Sub
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 ColNumb = 1 Then ColNumb = 8
350     'On Error Resume Next
351     If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
352     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
353         ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
354     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
355     ActiveCell.Offset(0, 1).Select
356 End Sub
357
358 Sub ShowSettings()
359     'Show or Hide Calendar Settings Panel
360     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
361         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
362             "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
363             = False
364     Else:
365         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
366             "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
367             = True
368     End If
369 End Sub

```

```
359 Sub UnGroupCal()  
360     On Error Resume Next  
361     ActiveSheet.Shapes("Calendar").Ungroup  
362     ActiveSheet.Shapes("NextMonth").Ungroup  
363     ActiveSheet.Shapes("PrevMonth").Ungroup  
364     On Error GoTo 0  
365 End Sub
```

—  
\_, 19

## A

Activate, 17  
ActiveCell, 7, 15, 16, 21  
ActiveSheet, 15-17, 19-22  
ActiveWindow, 10  
ActiveWorkbook, 16-18  
ActSht, 17  
Add, 17, 18  
Address, 16, 17  
Admin, 7, 12  
Admin\_SetTypeColor, 7  
AdvancedFilter, 12  
After, 17  
Application, 7, 8, 10, 12, 15, 21  
Appt\_CheckForMove, 10, 14  
Appt\_Delete, 8  
Appt\_Load, 8, 10, 14  
Appt\_New, 8  
Appt\_SaveUpdate, 8  
ApptCol, 8, 10, 12, 14  
ApptColor, 10, 12  
ApptDate, 10, 12  
ApptDur, 10, 12  
ApptID, 10, 12, 14  
ApptLeft, 10  
ApptName, 10, 12  
ApptRow, 8, 10, 12, 14  
ApptsDB, 8, 12, 14  
ApptShp, 10-12  
ApptTime, 10, 12  
ApptType, 10, 12  
Array, 15, 16, 19-21  
AutoFill, 18

## B

Bold, 15, 16, 20, 21

## C

Cal, 4  
CalCol, 15, 17  
Calendar, 2, 8, 10-14  
Calendar\_Appt\_Select, 10  
Calendar\_ChangeView, 2, 10  
Calendar\_GoToMonth, 10-12  
Calendar\_GoToWeek, 10, 11, 13  
Calendar\_NextMonth, 11  
Calendar\_NextWeek, 11  
Calendar\_PrevMonth, 11  
Calendar\_PrevWeek, 11  
Calendar\_Refresh, 8, 11, 12, 14  
Calendar\_ThisMonth, 12  
Calendar\_ThisWeek, 13  
CalendarHide, 4, 15  
CalendarShow, 4, 15  
CalFormulaReplacement, 16  
Caller, 7, 10, 15, 21  
CalRow, 10  
Cells, 8, 12, 14, 16, 17, 21  
CheckForSheet, 4, 16  
Chr, 2, 17, 18  
ClearContents, 8  
ColCnt, 17  
ColNum, 16  
ColNumb, 21  
Color, 7, 12  
Column, 14

Copy, 17, 21  
CopyToRange, 12  
Count, 17  
CountA, 8  
CountDelay, 14  
CreateCalSht, 17  
CreateWS, 16, 17  
CriteriaRange, 12

## D

DatePart, 10  
DateSerial, 10  
Day, 16  
DayCnt, 17  
DayName, 15, 16, 20  
DayNum, 15, 16  
DayNumb, 21  
Delete, 8, 11  
DestCol, 14  
DestDate, 14  
Destination, 18  
DestRow, 14  
DestTime, 14  
DoEvents, 14  
DrawingObject, 16, 17  
Duplicate, 12  
Duration, 12  
DurCols, 10, 12

## E

Empty, 2, 8, 14, 15, 20, 21  
EndDragDrop, 14  
EntireRow, 8, 10  
Explicit, 2-8, 10, 14

## F

Fill, 7, 12, 15, 16, 20  
Find, 12  
Font, 15, 16, 20, 21  
ForeColor, 7, 12, 15, 16, 20  
Format, 12  
Formula, 16, 17

## G

Group, 19  
GroupCal, 18, 19

## H

Height, 12

## I

InStr, 11, 16, 20  
Int, 12  
Interior, 7, 12  
Intersect, 2, 4  
IsDate, 15

## L

LastResultRow, 10, 12  
LastRow, 10, 12  
Left, 2, 10, 12, 14, 16, 20  
Len, 20

## M

MacroLink, 19, 20  
MacroLinkRemover, 16, 19  
Month, 12, 15, 17  
MonthStart, 10  
MsgBox, 8, 14, 16, 19  
msoBringToFront, 16

msoCTrue, 2, 10, 16, 19  
msoFalse, 2, 7, 10, 15, 20, 21  
msoTrue, 16

## N

Name, 11, 12, 17-19  
Names, 18  
NewLink, 19, 20  
NextAppt, 12  
NextMonth, 20  
NextShp, 20  
NextYear, 20  
NoCal, 15, 16, 19  
NotSaved, 8  
NumberFormat, 2, 18

## O

Offset, 16, 21  
OnAction, 12, 17, 20

## P

Paste, 17, 21  
Placement, 16, 19  
PrevMonth, 20  
PrevYear, 21

## R

Range, 2, 4, 8, 10-21  
RefersTo, 18  
Replace, 10, 21  
ReplaceCalendar, 4, 21  
ResultRow, 10, 12  
RGB, 7, 12, 15, 16, 20  
Right, 20  
RoundUp, 21  
Row, 2, 8, 12, 14  
RowCnt, 17  
RowHeight, 10  
RowNum, 16  
RowNumb, 21  
Rows, 10

## S

SchedDur, 10, 12  
SchedStart, 10, 12  
ScreenUpdating, 8, 12  
ScrollRow, 10  
SelCell, 15, 16  
SelectDay, 21  
Selection, 15, 19  
Shape, 4, 10, 16, 19  
ShapeRange, 15, 19  
Shapes, 2, 4, 7, 10-12, 14-17, 19-22  
Sheets, 15-17, 20, 21  
ShowSettings, 21  
Shp, 16, 19, 20  
Split, 20  
SplitLink, 19, 20  
StartDate, 10, 12  
StartTime, 12  
StartYearDate, 12

## T

Target, 2, 4  
Text, 12  
TextFrame2, 12, 15, 16, 20, 21  
TextRange, 12, 15, 16, 20, 21  
ThisWorkbook, 17, 20, 21  
Top, 2, 10, 12, 14, 16  
TopLeftCell, 14

TypeRow, [10](#), [12](#)

## U

Ungroup, [22](#)

UnGroupCal, [17](#), [22](#)

Unique, [12](#)

## V

Value, [2](#), [8](#), [10-18](#), [20](#), [21](#)

vbNo, [8](#)

vbYesNo, [8](#)

ViewType, [10](#)

Visible, [2](#), [4](#), [7](#), [10](#), [15-17](#), [19-21](#)

## W

WeekStart, [10](#)

Width, [12](#)

Worksheet, [16](#), [17](#)

Worksheet\_Change, [2](#)

Worksheet\_SelectionChange, [2](#), [4](#)

WorksheetFunction, [8](#), [21](#)

ws, [16](#), [17](#)

## X

xlFillDefault, [18](#)

xlFilterCopy, [12](#)

xlMove, [16](#), [19](#)

xlSheetHidden, [17](#)

xlUp, [8](#), [12](#)

xlValues, [12](#)

xlWhole, [12](#)

## Y

Year, [11](#), [12](#), [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*