

ABOUT THE AUTHOR

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

With over 442,000 YouTube subscribers, 33,000,000 video views, 250+ comprehensive training videos, and a thriving 60,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

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



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



Microsoft®
Most Valuable
Professional



OUR COURSES & PRODUCTS



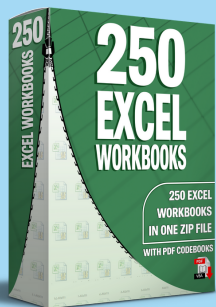
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



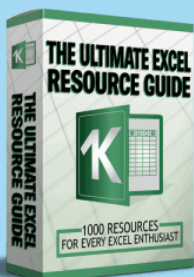
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



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

[Click here to learn more](#)



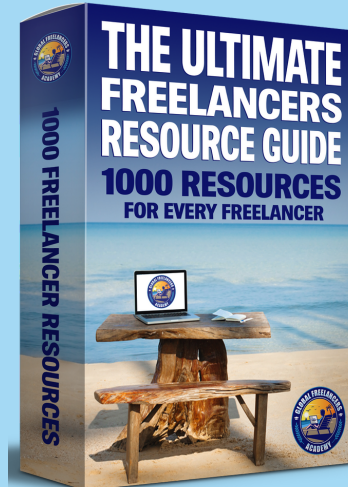
With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)



This incredible 13-hour freelancer masterclass will teach anyone how to be a successful freelancer with my proven 9-Phase 'Financial Freedom Roadmap' and includes 30+ downloads and exercises along with a ton of resources for anyone to reach their goals through freelancing.

[Click here to learn more](#)



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

[Click here to learn more](#)

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Worksheet_SelectionChange [Sub].....	2
Schedule.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub].....	3
Sheet1.....	4
(Declarations).....	4
SMPosts.....	5
(Declarations).....	5
ThisWorkbook.....	6
(Declarations).....	6
Modules.....	7
Application_Macros.....	7
(Declarations).....	7
Admin_SetPlatformColor [Sub].....	7
BrowseForIconFolder [Sub].....	7
BrowseForIconPicture [Sub].....	7
ChatGPT_Macros.....	8
(Declarations).....	8
GenerateAndSchedulePostSeries [Sub].....	8
GeneratePostContent [Sub].....	10
PopUp_Calendar.....	12
(Declarations).....	12
CalCol [Sub].....	12
CalendarHide [Sub].....	12
CalendarShow [Sub].....	12
CalFormulaReplacement [Sub].....	13
CheckForSheet [Sub].....	13
CreateCalSht [Sub].....	14
GroupCal [Sub].....	16
MacroLinkRemover [Sub].....	16
NextMonth [Sub].....	17
NextYear [Sub].....	17
PrevMonth [Sub].....	17
PrevYear [Sub].....	18
ReplaceCalendar [Sub].....	18
SelectDay [Sub].....	18
ShowSettings [Sub].....	18
UnGroupCal [Sub].....	18
Schedule_Macros.....	19
(Declarations).....	19
Sched_DayLoad [Sub].....	19
Sched_NextMonth [Sub].....	20
Sched_Post_Select [Sub].....	20
Sched_PostLoad [Sub].....	20
Sched_PrevMonth [Sub].....	21
Sched_Refresh [Sub].....	21
Sched_ThisMonth [Sub].....	23
SMPost_Macros.....	24
(Declarations).....	24
Post_AddMedia [Sub].....	24
Post_Delete [Sub].....	24
Post_NewSeries [Sub].....	24
Post_NewSinglePost [Sub].....	24
Post_SaveUpdate [Sub].....	24

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5     If Shapes("ColorPalette" ).Visible = True Then Shapes("ColorPalette" ).Visible =
6     msoFalse
7     If Shapes("ColorPalette" ).Visible = True Then Shapes("BrowseIconBtn" ).Visible =
8     msoFalse
9
10    'Show color palette
11    If Not Intersect(Target, Range("R6:R25" )) Is Nothing And Range("Q" & Target.Row).
12    Value <> Empty Then
13        With Shapes("ColorPalette" )
14            .Left = Range("R" & Target.Row).Left
15            .Top = Range("R" & Target.Row + 1).Top
16            .Visible = msoCTrue
17        End With
18    End If
19
20    'Show Browse For Platform Icon
21    If Not Intersect(Target, Range("Q6:S25" )) Is Nothing And Range("Q" & Target.Row).
22    Value <> Empty Then
23        With Shapes("BrowseIconBtn" )
24            .Left = Range("T" & Target.Row).Left
25            .Top = Range("T" & Target.Row).Top
26            .Visible = msoCTrue
27        End With
28    End If
29 End Sub
```

```
1 Option Explicit
2
3
4
5 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
6     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
       it in
7     Dim Cal As Shape
8     On Error Resume Next
9     Set Cal = Shapes("Calendar" )
10    On Error GoTo 0
11    If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
12    If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
13
14    If Target.Count > 1 Then Exit Sub
15    If Not Intersect(Target, Range("F5,H5,P10" )) Is Nothing Then 'Change this to any
       cell (or cells) you would like to have the Pop-Up Calendar Appear
16        CheckForSheet
17        CalendarShow
18    Else:
19        CheckForSheet
20        CalendarHide
21    End If
22 End Sub
```

1
2

Option Explicit

1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2
3 Sub Admin_SetPlatformColor()
4     ActiveCell.Interior.Color = Admin.Shapes(Application.Caller).Fill.ForeColor.RGB
5     Admin.Shapes("ColorPalette").Visible = msoFalse
6 End Sub
7
8
9 Sub BrowseForIconFolder()
10     Dim BillFolder As FileDialog
11     Set BillFolder = Application.FileDialog(msoFileDialogFolderPicker)
12     With BillFolder
13         .Title = "Browse For Icon Folder"
14         .AllowMultiSelect = False
15         If .Show <> -1 Then GoTo NoSelection
16         Admin.Range("D5").Value = .SelectedItems(1)
17     End With
18     NoSelection:
19 End Sub
20
21 Sub BrowseForIconPicture()
22     Dim IconFile As FileDialog
23     If [IconFolder] = "" Or Dir([IconFolder], vbDirectory) = "" Then
24         MsgBox "Please select a correct folder for the icons"
25         BrowseForIconFolder
26     End If
27
28     Set IconFile = Application.FileDialog(msoFileDialogFilePicker)
29     With IconFile
30         .Title = "Browse For Platform Icon"
31         .AllowMultiSelect = False
32         .Filters.Add "Pictures Files", "*.jpg, *.png, *.gif, *.jpeg, *.GIF", 1
33         If .Show <> -1 Then GoTo NoSelection
34         Admin.Range("S" & ActiveCell.Row).Value = Dir(.SelectedItems(1)) 'Set Icon Name
35
36         If .SelectedItems(1) = [IconFolder] & "\" & Dir(.SelectedItems(1)) Then Exit Sub
37         'Icon already exists in icon folder
38         FileCopy .SelectedItems(1), [IconFolder] & "\" & Dir(.SelectedItems(1))
39     End With
40     NoSelection:
41 End Sub
42
43
```

```

1 Option Explicit
2 Dim oXMLHTTP As Object
3 Dim AIBody As String, Response As String, APIKey As String, Prompt As String, Temp As
String, Model As String
4 Dim Product As String, ProdDesc As String, Platform As String, ParsedAnswer As String,
Frequency As String
5 Dim ProdRow As Long, PostRow As Long, PostCount As Long
6 Dim PostDate As Date, StartOn As Date, EndOn As Date, PostAt As Date
7 Sub GenerateAndSchedulePostSeries()
8     With Schedule
9         'Check For Required Fields (all 6)
10        If .Range("B14" ).Value < 6 Then
11            MsgBox "Please make sure to fill in all 6 required fields"
12            Exit Sub
13        End If
14        If .Range("B16" ).Value = Empty Then
15            MsgBox "Please select a correct product from the drop down list"
16            .Range("H3" ).Select
17            Exit Sub
18        End If
19
20        APIKey = Admin.Range("G28" ).Value 'Secret Key
21        Model = Admin.Range("G29" ).Value 'Model
22        Temp = Admin.Range("G30" ).Value 'Temperature
23        If Model = "" Then Model = "text-davinci-003" 'Set Default Model
24        If Temp = "" Then Temp = "0.4" 'Set Default Temp
25        PostCount = 0 'Set Intial Post Count
26        Platform = .Range("H3" ).Value 'Plaform
27        Product = .Range("F3" ).Value ' Product
28        PostAt = .Range("J3" ).Value 'Post At Time
29        StartOn = .Range("F5" ).Value 'Start On Date
30        PostDate = StartOn 'Set Initial Posting Date
31        EndOn = .Range("H5" ).Value 'End On Date
32        Frequency = .Range("J5" ).Value 'Frequency
33        ProdRow = .Range("B16" ).Value 'Product Row
34        ProdDesc = Admin.Range("G" & ProdRow).Value 'Product Description
35        Prompt = "Please create a social media post for " & ProdDesc & " for the " &
Platform & " platform. Include hashtags"
36        Prompt = Replace(Replace(Prompt, Chr(34), "\" & Chr(34)), vbNewLine, "\n" )
37        AIBody = "{" "model" ": " & Model & " ", "prompt" ": " & Prompt & "
", " "temperature" ": " & Temp & " ", " "max_tokens" ": 3500}" 'Request Body
38        For Text Response
39        Debug.Print AIBody
40    Do While PostDate <= EndOn 'Begin Loop
41        'On Error GoTo TimeOut
42        Set oXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP" )
43        oXMLHTTP.Open "POST" , "https://api.openai.com/v1/completions" , False 'Set
Open Posts For Text
44        oXMLHTTP.setRequestHeader "Content-Type" , "application/json" 'Set Header
Type
45        oXMLHTTP.setRequestHeader "Authorization" , "Bearer " & APIKey 'Set Header
Authorization
46        oXMLHTTP.Send AIBody
47        Response = oXMLHTTP.responseText 'Extract Response
48        If InStr(Response, "You didn't provide an API key" ) > 0 Then
49            MsgBox "No API Key was added. You can find your API key at " & vbCrLf &
"https://platform.openai.com/account/api-keys."
50            Exit Sub
51        End If
52        If InStr(Response, "Incorrect API key provided" ) > 0 Then

```

1 2 3 4

```

53   1   2   3   4   MsgBox "Incorrect API key provided. You can find your API key at " &
    vbCrLf & "https://platform.openai.com/account/api-keys."
54   Exit Sub
55   End If
56   If InStr(Response, "maximum context length" ) > 0 Then
57     MsgBox "You have reached the maximum content length. Please clear the
    history above to move forward"
58     Exit Sub
59   End If
60   If InStr(Response, "exceeded your current quota" ) > 0 Then
61     MsgBox "You have exceeded your current ChatGPT API Quota. Please consider
    upgrading your account"
62     Exit Sub
63   End If
64   Debug.Print Response
65   ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "{" & Chr(
    34) & "text" & Chr(34) & ":" & Chr(34)) - 9)
66   ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(
    34) & "index" ) - 1)
67   ParsedAnswer = Replace(Replace(ParsedAnswer, "\n" , Chr(10)), "\" & Chr(34),
    Chr(34))
68   If Left(ParsedAnswer, 1) = "." Then ParsedAnswer = Right(ParsedAnswer, Len(
    ParsedAnswer) - 1)
69   If Left(ParsedAnswer, 2) = Chr(10) & Chr(10) Then ParsedAnswer = Right(
    ParsedAnswer, Len(ParsedAnswer) - 2) 'Remove any initial carriage returns
70
71   With SMPosts
72     PostRow = .Range("A99999" ).End(xlUp).Row + 1 'First Avail. Row
73     .Range("A" & PostRow).Value = Schedule.Range("B12" ).Value 'Set Next Post
    ID
74     .Range("B" & PostRow).Value = Platform 'Social Media Platform
75     .Range("C" & PostRow).Value = Product 'Product/Service
76     .Range("D" & PostRow).Value = PostDate 'Post on Date
77     .Range("E" & PostRow).Value = PostAt 'Post At Time
78     .Range("G" & PostRow).Value = "Pending" 'Set Pending
79     .Range("H" & PostRow).Value = ParsedAnswer 'Post Details/ Text
80     .Range(PostRow & ":" & PostRow).WrapText = False 'Contain Row height
81   End With
82   'Update Next Post Date Based on Frequency
83   Select Case Frequency
84     Case Is = "Every Day"
85       PostDate = PostDate + 1 'Increment 1 Day
86     Case Is = "Every Weekday" 'Non Weekends
87       PostDate = Application.WorksheetFunction.WorkDay(PostDate, 1) 'Exclude
    Weekends
88     Case Is = "Weekly"
89       PostDate = PostDate + 7 'Increment 1 Week
90     Case Is = "Monthly"
91       PostDate = DateAdd("m" , 1, PostDate) 'Increment 1 month
92   End Select
93   PostCount = PostCount + 1
94   ParsedAnswer = ""
95   Response = ""
96   Loop
97   End With
98   Sched_Refresh 'Refresh Schedule
99   Sched_DayLoad 'Refresh Day View
100  MsgBox PostCount & " posts have been created & added to the schedule"
101  Exit Sub
102  Timeout:
103  MsgBox "The request has timed out. Please try requesting a smaller set of data"

```

1

```

1
104 End Sub
105
106
107
108
109 Sub GeneratePostContent()
110 With Schedule
111 'Check For Correct Platform & Product
112 If .Range("P8").Value = "" Or .Range("P9").Value = "" Or .Range("B15").Value
113 = Empty Then
114     MsgBox "Please select a platform and a product from the drop down lists"
115     .Range("P8").Select
116     Exit Sub
117 End If
118 APIKey = Admin.Range("G28").Value 'Secret Key
119 Model = Admin.Range("G29").Value 'Model
120 Temp = Admin.Range("G30").Value 'Temperature
121 If Model = "" Then Model = "text-davinci-003" 'Set Default Model
122 If Temp = "" Then Temp = "0.4" 'Set Default Temp
123 Platform = .Range("P8").Value 'Plaform
124 Product = .Range("P9").Value ' Product
125 ProdRow = .Range("B15").Value 'Product Row
126 ProdDesc = Admin.Range("G" & ProdRow).Value 'Product Description
127 Prompt = "Please create a social media post for " & ProdDesc & " for the " &
128 Platform & " platform. Include hashtags"
129 Prompt = Replace(Replace(Prompt, Chr(34), "\" & Chr(34)), vbNewLine, "\n" )
130 Set oXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP" )
131 oXMLHTTP.Open "POST", "https://api.openai.com/v1/completions", False 'Set Open
132 Posts For Text
133 AIBody = "{" "model" ": " "" & Model & " ", " "prompt" ": " "" & Prompt & " "
134 ", " "temperature" ": " & Temp & " ", " "max_tokens" ": 3500}" 'Request Body
135 For Text Response
136 oXMLHTTP.setRequestHeader "Content-Type", "application/json" 'Set Header Type
137 oXMLHTTP.setRequestHeader "Authorization", "Bearer " & APIKey 'Set Header
138 Authorization
139 Debug.Print AIBody
140 On Error GoTo TimeOut
141 oXMLHTTP.Send AIBody
142 Response = oXMLHTTP.responseText 'Extract Response
143 If InStr(Response, "You didn't provide an API key" ) > 0 Then
144     MsgBox "No API Key was added. You can find your API key at " & vbCrLf &
145     "https://platform.openai.com/account/api-keys."
146     Exit Sub
147 End If
148 If InStr(Response, "Incorrect API key provided" ) > 0 Then
149     MsgBox "Incorrect API key provided. You can find your API key at " & vbCrLf &
150     "https://platform.openai.com/account/api-keys."
151     Exit Sub
152 End If
153 If InStr(Response, "maximum context length" ) > 0 Then
154     MsgBox "You have reached the maximum content length. Please clear the history
155     above to move forward"
156     Exit Sub
157 End If
158 If InStr(Response, "exceeded your current quota" ) > 0 Then
159     MsgBox "You have exceeded your current ChatGPT API Quota. Please consider
160     upgrading your account"
161     Exit Sub
162 End If
163 Debug.Print Response

```

1 2

```
155   1 2   ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "{" & Chr(34)
      & "text" & Chr(34) & ":" & Chr(34)) - 9)
156   ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(34)
      & "index" ) - 1)
157   ParsedAnswer = Replace(Replace(ParsedAnswer, "\n" , Chr(10)), "\" & Chr(34), Chr
      (34))
158   If Left(ParsedAnswer, 1) = "." Then ParsedAnswer = Right(ParsedAnswer, Len(
      ParsedAnswer) - 1)
159   If Left(ParsedAnswer, 2) = Chr(10) & Chr(10) Then ParsedAnswer = Right(
      ParsedAnswer, Len(ParsedAnswer) - 2) 'Remove any initial carriage returns
160   .Range("O15" ).Value = ParsedAnswer
161   End With
162   Exit Sub
163   TimeOut:
164   MsgBox "The request has timed out. Please try requesting a smaller set of data"
165   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
58         If InStr(.Shapes("1Day" ).DrawingObject.Formula, "]" ) <> 0 Then 'Run Workbook

```

1 2 3


```

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

```

```

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

```

1 2

```

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

```

```

210 Sub GroupCal()
211     ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
212     Selection.ShapeRange.Name = "NextMonth"
213     ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
214     Selection.ShapeRange.Name = "PrevMonth"
215     ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
216         , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" ,
217         "1Day" , _
218         "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" ,
219         "6Day" , _
220         "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
221         "18Day" _
222         , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day"
223         , _
224         "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
225         "Sa" , _
226         "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
227         "CalCol3" , _
228         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" ,
229         "PrevMonth" , _
230         "NextMonth" , "NextYr" , "PrevYr" )).Visible = msoCTrue
231     ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
232         , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day"
233         , _
234         "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
235         "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
236         "18Day" _
237         , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
238         "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
239         "Sa" , _
240         "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
241         "CalCol3" , _
242         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth"
243         , _
244         "NextMonth" , "NextYr" , "PrevYr" )).Select
245     Selection.ShapeRange.Group.Select
246     Selection.ShapeRange.Name = "Calendar"
247     Selection.Name = "Calendar"
248     Selection.Placement = xlMove
249     ActiveSheet.Shapes("Calendar" ).Placement = 2
250 End Sub

```

```

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

```

```

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

```

1 2 3

```

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

```

```

1 Option Explicit
2 Dim PostRow As Long, PostCol As Long, LastRow As Long, LastResultRow As Long, ResultRow
  As Long, SchedRow As Long, SchedCol As Long
3 Dim CalRow As Long, CalCol As Long, PlatformRow As Long, DayPostCount As Long, PostNumb
  As Long
4 Dim PlatformColor As String, PostID As String, Platform As String, PostText As String
5 Dim IconPic As String, PicPath As String, GrpString As String, GrpArr() As String
6 Dim PostShp As Shape
7 Dim PostDate As Date, PostTime As Date
8 Dim PostHeight As Double, PostWidth As Double, PostLeft As Double, PostTop As Double
9 Sub Sched_DayLoad()
10     With SMPosts
11         For Each PostShp In Schedule.Shapes
12             On Error Resume Next
13             If InStr(PostShp.Name, "Day_Post" ) > 0 Then PostShp.Delete 'Delete Day Post
  Shape
14             On Error GoTo 0
15         Next PostShp
16         LastRow = .Range("A99999" ).End(xlUp).Row 'Last Row
17         If LastRow < 4 Then Exit Sub
18         .Range("Q3:X99999" ).ClearContents 'Clear Previous Result
19         'Show all posts for the selected date
20         .Range("A3:H" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
  "O2:O3" ), CopyToRange:=.Range("Q2:X2" ), Unique:=True
21         LastResultRow = .Range("Q99999" ).End(xlUp).Row
22         If LastResultRow < 3 Then Exit Sub
23         If LastResultRow < 4 Then GoTo NoSort
24         With .Sort 'Sort Based On Post Time
25             .SortFields.Clear
26             .SortFields.Add Key:=SMPosts.Range("U3" ), SortOn:=xlSortOnValues, Order:=
  xlAscending, DataOption:=xlSortNormal 'Sort By Post Time
27             .SetRange SMPosts.Range("Q3:X" & LastResultRow)
28             .Apply
29         End With
30         NoSort:
31         PostTop = Schedule.Range("M8" ).Top + 2 'Set Intial Top Position
32         PostWidth = Schedule.Range("M:M" ).Width 'Set Post Shape Width
33         PostHeight = Schedule.Range("8:13" ).Height 'Height equal to 6 Rows
34         PostLeft = Schedule.Range("M1" ).Left 'Set Left Postion
35         For ResultRow = 3 To LastResultRow
36             PostID = .Range("Q" & ResultRow).Value 'Post ID
37             PostDate = .Range("T" & ResultRow).Value 'Post Date
38             PostTime = .Range("U" & ResultRow).Value 'Post. Time
39             Platform = .Range("R" & ResultRow).Value 'Platform
40             PostText = .Range("X" & ResultRow).Value 'Post Status
41
42             On Error Resume Next
43             PlatformRow = Admin.Range("Platforms" ).Find(Platform, , xlValues, xlWhole).
  Row 'Get Platform Row
44             On Error GoTo 0
45             If PlatformRow <> 0 Then
46                 PlatformColor = Admin.Range("R" & PlatformRow).Interior.Color 'Add Cell
  Color into Variable
47                 IconPic = Admin.Range("S" & PlatformRow).Value 'Platform Picture
48             End If
49             Schedule.Shapes("SamplePostDay" ).Duplicate.Name = "Day_Post" & PostID
50             With Schedule.Shapes("Day_Post" & PostID)
51                 .Left = PostLeft 'Set Left Position
52                 .Top = PostTop 'Set Top Position
53                 .Width = PostWidth 'Set Width
54                 .Height = PostHeight 'Set Height Position

```

1 2 3 4


```

55     1 2 3 4     .TextFrame2.TextRange.Text = Format(PostTime, "H:MMa/p" ) & " " & PostText
                    'Time & first 10 characters of post text
56     .OnAction = "Sched_PostLoad" 'Assign macro to shape
57     If PlatformColor <> 0 Then .Fill.ForeColor.RGB = PlatformColor 'Set Shape
                    color
58     End With
59
60     'Add in Platform Icon Picture
61     PicPath = [IconFolder] & "\" & IconPic 'Platform Picture File Path
62     Schedule.Shapes("SampleIconShp" ).Duplicate.Name = "Day_PostIcon" & PostID
                    'Create Shape for Platform Picture
63     With Schedule.Shapes("Day_PostIcon" & PostID)
64         If Dir(PicPath, vbDirectory) <> "" And IconPic <> "" Then .Fill.
                    UserPicture (PicPath) 'Add In Platform Picture
65         .Left = PostLeft + PostWidth - 15
66         .Top = PostTop + 2
67     End With
68     PostTop = PostTop + PostHeight + 3 'Reset Top Position
69     Next ResultRow
70 End With
71 End Sub

```

```

72
73 Sub Sched_NextMonth()
74     With Schedule
75         If .Range("B2" ).Value = 9 And .Range("B7" ).Value = 12 Then
76             MsgBox "You are at the last month and the last year of the schedule"
77             Exit Sub
78         End If
79         If .Range("B7" ).Value = 12 Then 'On December
80             .Range("B2" ).Value = .Range("B2" ).Value + 1 'Increment the year up 1
81             .Range("B7" ).Value = 1 'Set Month to January
82         Else 'Not December
83             .Range("B7" ).Value = .Range("B7" ).Value + 1 'Increment month up 1
84         End If
85         Sched_Refresh 'Run Macro to refresh schedule
86     End With
87 End Sub

```

```

88
89 Sub Sched_Post_Select()
90     With Schedule
91         PostID = Replace(Application.Caller, Left(Application.Caller, 8), "" ) 'Extract
                    Post ID
92         Schedule.Range("B10" ).Value = PostID
93         PostRow = .Range("B11" ).Value 'Post Row
94         If PostRow = 0 Then Exit Sub
95         PostDate = SMPosts.Range("D" & PostRow).Value 'Post Date
96         .Range("M7" ).Value = PostDate 'Set Post Date
97         Sched_DayLoad 'Load Selected Days Posts
98         Sched_PostLoad 'Run Macro to display selected post
99     End With
100 End Sub

```

```

101 Sub Sched_PostLoad()
102     With Schedule
103         PostID = Replace(Application.Caller, Left(Application.Caller, 8), "" )
104         Schedule.Range("B10" ).Value = PostID
105         .Range("P8:P13,015:P43" ).ClearContents 'Clear Selected Post Details
106         PostRow = .Range("B11" ).Value
107         If PostRow = 0 Then Exit Sub
108         For PostCol = 2 To 7
109             .Range("P" & PostCol + 6).Value = SMPosts.Cells(PostRow, PostCol).Value '
110         Next PostCol

```



```

1 2
111 .Range("O15" ).Value = SMPosts.Range("H" & PostRow).Value 'Post Details
112 End With
113 End Sub
114
115 Sub Sched_PrevMonth()
116 With Schedule
117 If .Range("B2" ).Value = 1 And .Range("B7" ).Value = 1 Then
118     MsgBox "You are at the first month and the first year of the schedule"
119     Exit Sub
120 End If
121 If .Range("B7" ).Value = 1 Then 'On January
122     .Range("B2" ).Value = .Range("B2" ).Value - 1 'Increment the year down 1
123     .Range("B7" ).Value = 12 'Set Month to December
124 Else 'Not January
125     .Range("B7" ).Value = .Range("B7" ).Value - 1 'Increment month down 1
126 End If
127 Sched_Refresh 'Run Macro to refresh schedule
128 End With
129 End Sub
130
131 Sub Sched_Refresh()
132 'Clear all existing Post shapes from the schedule
133 For Each PostShp In Schedule.Shapes
134     On Error Resume Next
135     If InStr(PostShp.Name, "PostItem" ) > 0 Then PostShp.Delete 'Delete Post Shape
136     If InStr(PostShp.Name, "PlatIcon" ) > 0 Then PostShp.Delete 'Delete Platform
137     Picture Shape
138     On Error GoTo 0
139 Next PostShp
140 With SMPosts
141     LastRow = .Range("A99999" ).End(xlUp).Row 'Last Row SM Post
142     If LastRow < 4 Then Exit Sub
143     .Range("Q3:X99999" ).ClearContents 'Clear Previous Result
144     .Range("A3:H" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
145     "M2:N3" ), CopyToRange:=.Range("Q2:X2" ), Unique:=True
146     LastResultRow = .Range("Q99999" ).End(xlUp).Row
147     If LastResultRow < 3 Then Exit Sub
148     Application.ScreenUpdating = False
149     If LastResultRow < 4 Then GoTo NoSort
150     With .Sort 'Sort Based On Post Date & time
151         .SortFields.Clear
152         .SortFields.Add Key:=SMPosts.Range("T3" ), SortOn:=xlSortOnValues, Order:=
153         xlAscending, DataOption:=xlSortNormal 'Sort By Post Date
154         .SortFields.Add Key:=SMPosts.Range("U3" ), SortOn:=xlSortOnValues, Order:=
155         xlAscending, DataOption:=xlSortNormal 'Sort By Post Time
156         .SetRange SMPosts.Range("Q3:X" & LastResultRow)
157         .Apply
158     End With
159     NoSort:
160     For ResultRow = 3 To LastResultRow
161         PostID = .Range("Q" & ResultRow).Value 'Post ID
162         PostDate = .Range("T" & ResultRow).Value 'Post Date
163         PostTime = .Range("U" & ResultRow).Value 'Post. Time
164         Platform = .Range("R" & ResultRow).Value 'Platform
165         PostText = .Range("X" & ResultRow).Value 'Post Status
166         On Error Resume Next
167         PlatformRow = Admin.Range("Platforms" ).Find(Platform, , xlValues, xlWhole).
168         Row 'Get Platform Row
169         On Error GoTo 0

```

1 2 3

```

167 | 1 2 3 | If PlatformRow <> 0 Then 'Platfrom row found
168 | | PlatformColor = Admin.Range("R" & PlatformRow).Interior.Color 'Add Cell
    | | Color into Variable
169 | | IconPic = Admin.Range("S" & PlatformRow).Value 'Platform Icon Picture
170 | | End If
171 | | DayPostCount = Application.WorksheetFunction.CountIf([PostDate_Results],
    | | PostDate) 'Get # of Posts in a single day
172 | | If DayPostCount > 5 Then DayPostCount = 5 'Set Max Posts to a single day
173 | | If DayPostCount > 1 Then PostNumb = PostNumb + 1 Else PostNumb = 1 'Track
    | | Post Number
174 | | 'Locate calendar row and column based on Date (Loop through rows & columns)
175 | | If PostNumb = 1 Then
176 | |     For CalRow = 8 To 38 Step 6
177 | |         For CalCol = 5 To 11
178 | |             If Schedule.Cells(CalRow, CalCol).Value = PostDate Then
179 | |                 SchedRow = CalRow 'Set Row For Post
180 | |                 SchedCol = CalCol 'Set Col For Post.
181 | |                 GoTo CreatePost
182 | |             End If
183 | |         Next CalCol
184 | |     Next CalRow
185 | | Else 'Same day Post
186 | |     SchedRow = SchedRow + 1 'Increment Row on Same Day Posts
187 | | End If
188 | | CreatePost:
189 | | If SchedRow = 0 Or SchedCol = 0 Then GoTo NextPost
190 | | Schedule.Shapes("SamplePostShp" ).Duplicate.Name = "PostItem" & PostID
191 | | PostWidth = Schedule.Cells(SchedRow, SchedCol).Width 'Set Post Shape Width
192 | | PostLeft = Schedule.Cells(SchedRow, SchedCol).Left 'Set Left Postion
193 | |
194 | | With Schedule.Shapes("PostItem" & PostID)
195 | |     .Left = PostLeft 'Set Left Postion
196 | |     .Top = Schedule.Cells(SchedRow + 1, SchedCol).Top + 1
197 | |     .Width = PostWidth 'Set Shape Width
198 | |     .Height = 15
199 | |     .TextFrame2.TextRange.Text = Format(PostTime, "H:MMa/p" ) & ": " & Left(
    | |     PostText, 20) 'Time & first 20 characters of post text
200 | |     .OnAction = "Sched_Post_Select" 'Assign macro to shape
201 | |     If PlatformColor <> 0 Then .Fill.ForeColor.RGB = PlatformColor 'Set Shape
    | |     color
202 | | End With
203 | | GrpString = "PostItem" & PostID
204 | |
205 | | 'Add in Platform Icon Picture
206 | | PicPath = [IconFolder] & "\" & IconPic 'Platform Picture File Path
207 | | Schedule.Shapes("SampleIconShp" ).Duplicate.Name = "PlatIcon" & PostID
    | | 'Create Shape for Platform Picture
208 | | With Schedule.Shapes("PlatIcon" & PostID)
209 | |     If Dir(PicPath, vbDirectory) <> "" And IconPic <> "" Then .Fill.
    | |     UserPicture (PicPath) 'Add In Platform Picture
210 | |     .Left = PostLeft
211 | |     .Top = Schedule.Cells(SchedRow + 1, SchedCol).Top + 1
212 | |     .OnAction = "Sched_Post_Select" 'Assign macro to shape
213 | | End With
214 | | GrpString = GrpString & "," & "PlatIcon" & PostID 'String used to group
    | | Post shapes
215 | | GrpArr = Split(GrpString, ",") 'Create Array from String
216 | | Schedule.Shapes.Range(GrpArr).Group.Name = "PostItemGrp" & PostID 'Unique
    | | Group Name
217 | | 'Clean up variables
218 | | Platform = Empty 'Clear out Platform name

```

```
1 2 3
219 GrpString = "" 'Clear out Group string
220 PlatformRow = 0 ' Reset Status row
221 PlatformColor = 0 'Reset Status color
222 If PostNumb >= DayPostCount Then PostNumb = 0
223 NextPost:
224     Next ResultRow
225 End With
226 Application.ScreenUpdating = True
227 End Sub
228
229
230 Sub Sched_ThisMonth()
231     Schedule.Range("B7").Value = Month(Date) 'Set Current Month
232     On Error Resume Next
233     Schedule.Range("B2").Value = Schedule.Range("Years").Find(Year(Date), , xlFormulas
        , xlWhole).Row - 21
234     On Error GoTo 0
235     Sched_Refresh 'Run Macro to refresh schedule
236 End Sub
237
238
```

```

1 Option Explicit
2 Dim PostRow As Long, PostCol As Long
3 Sub Post_AddMedia()
4     Dim MediaFile As FileDialog
5     Set MediaFile = Application.FileDialog(msoFileDialogFilePicker)
6     With MediaFile
7         .Title = "Browse For Picture Or Video"
8         .AllowMultiSelect = False
9         .Filters.Add "Pictures or Video Files", "*.jpg, *.png, *.gif, *.jpeg, *.GIF,
10         *.mp4, *.mov,*.mp3" , 1
11         If .Show <> -1 Then GoTo NoSelection
12         Schedule.Range("P12" ).Value = .SelectedItems(1) 'Add full file path
13     End With
14     NoSelection:
15 End Sub
16 Sub Post_Delete()
17     If MsgBox("Are you sure you want to delete this post?" , vbYesNo, "Delete Post" ) =
18     vbNo Then Exit Sub
19     With Schedule
20         If .Range("B11" ).Value = Empty Then GoTo NotSaved
21         PostRow = .Range("B11" ).Value 'Post Row
22         SMPosts.Range(PostRow & ":" & PostRow).EntireRow.Delete
23         NotSaved:
24         Post_NewSinglePost 'Clear out post fields
25         Sched_Refresh 'Refresh Schedule
26         Sched_DayLoad 'Refresh Day View
27     End With
28 End Sub
29 Sub Post_NewSeries()
30     Schedule.Range("F3,H3,J3,F5,H5" ).ClearContents 'clear out all fields
31     Schedule.Range("F3" ).Select
32 End Sub
33 Sub Post_NewSinglePost()
34     Schedule.Range("B10,P8:P13,015:P43" ).ClearContents 'clear out all fields
35     Schedule.Range("P8" ).Select
36 End Sub
37 Sub Post_SaveUpdate()
38     With Schedule
39         If .Range("B13" ).Value < 4 Then
40             MsgBox "Please make sure to add in a Platform, Product/Service, Schedule Date
41             & Schedule Time before saving"
42             Exit Sub
43         End If
44         If .Range("B11" ).Value = Empty Then 'New Post.
45             PostRow = SMPosts.Range("A99999" ).End(xlUp).Row + 1 'First Avail. Row
46             .Range("B10" ).Value = .Range("B12" ).Value 'Set Post ID.
47             SMPosts.Range("A" & PostRow).Value = .Range("B10" ).Value 'Set Post ID
48         Else: 'Existing Post
49             PostRow = .Range("B11" ).Value 'Existing Post ID
50         End If
51         For PostCol = 2 To 7
52             SMPosts.Cells(PostRow, PostCol).Value = .Range("P" & PostCol + 6).Value 'Add
53             in Post Details
54         Next PostCol
55         SMPosts.Range("H" & PostRow).Value = .Range("O15" ).Value 'Save Post details
56         Sched_Refresh 'Refresh Schedule
57         Sched_DayLoad 'Refresh Day View
58     End With
59 End Sub

```

—
_, 16

A

Activate, 14
ActiveCell, 7, 12, 13, 18
ActiveSheet, 12-14, 16-18
ActiveWorkbook, 13-15
ActSht, 14
Add, 7, 14, 15, 19, 21, 24
Address, 13, 14
Admin, 7, 8, 10, 19, 21, 22
Admin_SetPlatformColor, 7
AdvancedFilter, 19, 21
After, 14
AlBody, 8, 10
AllowMultiSelect, 7, 24
APIKey, 8, 10
Application, 7, 9, 12, 18, 20-24
Apply, 19, 21
Array, 12, 13, 16-18
AutoFill, 14, 15

B

BillFolder, 7
Bold, 12, 13, 17
BrowseForIconFolder, 7
BrowseForIconPicture, 7

C

Cal, 3
CalCol, 12, 14, 19, 22
CalendarHide, 3, 12
CalendarShow, 3, 12
CalFormulaReplacement, 13
Caller, 7, 12, 18, 20
CalRow, 19, 22
Cells, 13, 14, 18, 20, 22, 24
CheckForSheet, 3, 13
Chr, 8-11, 14, 15
Clear, 19, 21
ClearContents, 19-21, 24
ColCnt, 14
ColNum, 13
ColNumb, 18
Color, 7, 19, 22
Copy, 14, 18
CopyToRange, 19, 21
Count, 3, 14
CountIf, 22
CountLarge, 2
CreateCalSht, 14
CreateObject, 8, 10
CreatePost, 22
CreateWS, 13
CriteriaRange, 19, 21

D

DataOption, 19, 21
DateAdd, 9
Day, 13
DayCnt, 14
DayName, 12, 13, 17
DayNum, 12, 13
DayNumb, 18
DayPostCount, 19, 22, 23
Debug, 8-10
Delete, 19, 21, 24
Destination, 14, 15

Dir, 7, 20, 22
DrawingObject, 12-14
Duplicate, 19, 20, 22

E

Empty, 2, 8, 10, 12, 17, 18, 22, 24
EndOn, 8
EntireRow, 24
Explicit, 2-8, 19, 24

F

FileCopy, 7
FileDialog, 7, 24
Fill, 7, 12, 13, 17, 20, 22
Filters, 7, 24
Find, 19, 21, 23
Font, 12, 13, 17
ForeColor, 7, 12, 13, 17, 20, 22
Format, 20, 22
Formula, 12-14
Frequency, 8, 9

G

GenerateAndSchedulePostSeries, 8
GeneratePostContent, 10
Group, 16, 22
GroupCal, 15, 16
GrpArr, 19, 22
GrpString, 19, 22, 23

H

Height, 19, 22

I

IconFile, 7
IconFolder, 7, 20, 22
IconPic, 19, 20, 22
InStr, 8-10, 12, 17, 19, 21
InStrRev, 9, 11
Interior, 7, 19, 22
Intersect, 2, 3
IsDate, 12

K

Key, 19, 21

L

LastResultRow, 19, 21
LastRow, 19, 21
Left, 2, 9, 11, 13, 17, 19, 20, 22
Len, 9, 11, 17

M

MacroLink, 16, 17
MacroLinkRemover, 13, 16
MediaFile, 24
Model, 8, 10
Month, 12, 14, 23
MsgBox, 7-11, 13, 15, 20, 21, 24
msoBringToFront, 13
msoCTrue, 2, 13, 16
msoFalse, 2, 7, 12, 17, 18
msoFileDialogFilePicker, 7, 24
msoFileDialogFolderPicker, 7
msoTrue, 13

N

Name, 14-16, 19-22
Names, 15
NewLink, 16, 17

NextMonth, 17
NextPost, 22, 23
NextShp, 16, 17
NextYear, 17
NoCal, 12, 13, 15
NoSelection, 7, 24
NoSort, 19, 21
NotSaved, 24
NumberFormat, 15

O

Offset, 13, 18
OnAction, 14, 16, 17, 20, 22
Open, 8, 10
Order, 19, 21
oXMLHTTP, 8, 10

P

ParsedAnswer, 8, 9, 11
Paste, 14, 18
PicPath, 19, 20, 22
Placement, 13, 16
Platform, 8-10, 19, 21, 22
PlatformColor, 19, 20, 22, 23
PlatformRow, 19, 21-23
Post_AddMedia, 24
Post_Delete, 24
Post_NewSeries, 24
Post_NewSinglePost, 24
Post_SaveUpdate, 24
PostAt, 8, 9
PostCol, 19, 20, 24
PostCount, 8, 9
PostDate, 8, 9, 19-22
PostDate_Results, 22
PostHeight, 19, 20
PostID, 19-22
PostLeft, 19, 20, 22
PostNumb, 19, 22, 23
PostRow, 8, 9, 19-21, 24
PostShp, 19, 21
PostText, 19-22
PostTime, 19-22
PostTop, 19, 20
PostWidth, 19, 20, 22
PrevMonth, 17
PrevYear, 18
Print, 8-10
ProdDesc, 8, 10
ProdRow, 8, 10
Product, 8-10
Prompt, 8, 10

R

Range, 2, 3, 7-24
RefersTo, 15
Replace, 8-11, 18, 20
ReplaceCalendar, 3, 18
Response, 8-11
responseText, 8, 10
ResultRow, 19-21, 23
RGB, 7, 12, 13, 17, 20, 22
Right, 9, 11, 17
RoundUp, 18
Row, 2, 7, 9, 19, 21, 23, 24
RowCnt, 14
RowNum, 13
RowNumb, 18

S

Sched_DayLoad, 9, 19, 20, 24
Sched_NextMonth, 20
Sched_Post_Select, 20
Sched_PostLoad, 20
Sched_PrevMonth, 21
Sched_Refresh, 9, 20, 21, 23, 24
Sched_ThisMonth, 23
SchedCol, 19, 22
SchedRow, 19, 22
Schedule, 8-10, 19-24
ScreenUpdating, 21, 23
SelCell, 12, 13
SelectDay, 18
SelectedItems, 7, 24
Selection, 12, 16
Send, 8, 10
SetRange, 19, 21
SetRequestHeader, 8, 10
Shape, 3, 13, 16, 19
ShapeRange, 12, 16
Shapes, 2, 3, 7, 12-14, 16-22
Sheets, 12-14, 17, 18
Show, 7, 24
ShowSettings, 18
Shp, 13, 16, 17
SMPosts, 9, 19-21, 24
Sort, 19, 21
SortFields, 19, 21
SortOn, 19, 21
Split, 17, 22
SplitLink, 16, 17
StartOn, 8

T

Target, 2, 3
Temp, 8, 10
Text, 20, 22
TextFrame2, 12, 13, 17, 20, 22
TextRange, 12, 13, 17, 20, 22
ThisWorkbook, 14, 17, 18
TimeOut, 9-11
Title, 7, 24
Top, 2, 13, 19, 20, 22

U

Ungroup, 18
UnGroupCal, 14, 18
Unique, 19, 21
UserPicture, 20, 22

V

Value, 2, 7-15, 17-24
vbCrLf, 8-10
vbDirectory, 7, 20, 22
vbNewLine, 8, 10
vbNo, 24
vbYesNo, 24
Visible, 2, 3, 7, 12-14, 16-18

W

Width, 19, 22
WorkDay, 9
Worksheet, 13, 14
Worksheet_SelectionChange, 2, 3
WorksheetFunction, 9, 18, 22
WrapText, 9
ws, 13, 14

X

xlAscending, 19, 21
xlFillDefault, 14, 15
xlFilterCopy, 19, 21
xlFormulas, 23
xlMove, 13, 16
xlSheetHidden, 14
xlSortNormal, 19, 21
xlSortOnValues, 19, 21
xlUp, 9, 19, 21, 24
xlValues, 19, 21
xlWhole, 19, 21, 23

Y

Year, 12, 14, 23

Z

ZOrder, 13

Thank You!

This source code was created and made available to help you gain a better understanding of how VBA is used to create amazing Excel-based applications.

Thank you so much for your continued shares, likes and support. It really helps.



Excel For Freelancers