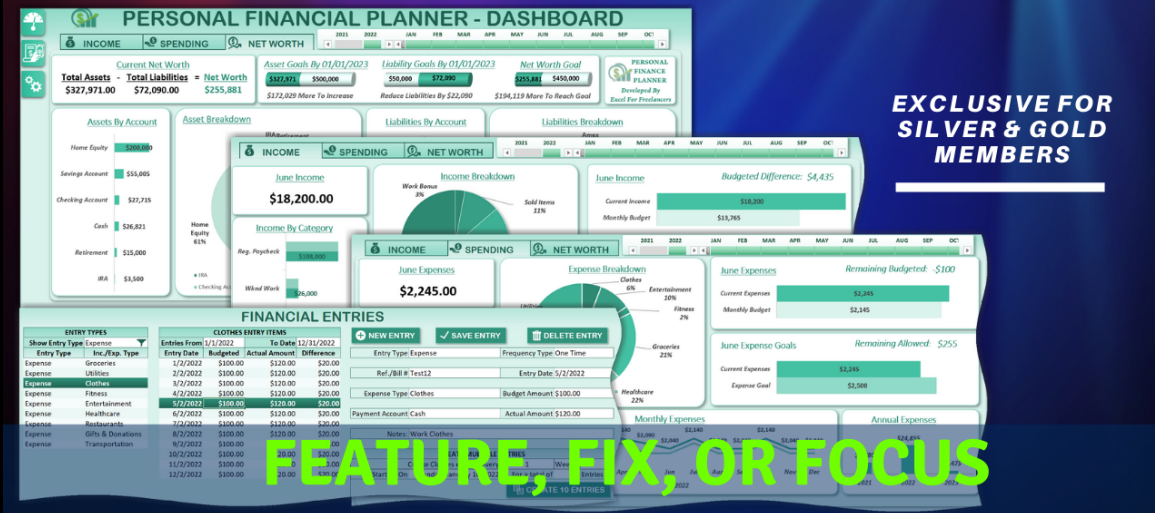


VBA SOURCE CODE BOOK

PERSONAL FINANCIAL PLANNER - UPDATED



EXCLUSIVE FOR
SILVER & GOLD
MEMBERS

FEATURE FIX, OR FOCUS



Excel For Freelancers

How To Create Your Own Personal Financial Planner



[DOWNLOAD
APPLICATION](#)



[VIEW
TRAINING](#)

by: Randy Austin

ABOUT THE AUTHOR

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

With over 300,000 YouTube subscribers, 15,000,000 video views, 200+ comprehensive training videos, and a thriving 50,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

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



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



Microsoft[®]
Most Valuable
Professional



OUR COURSES & PRODUCTS



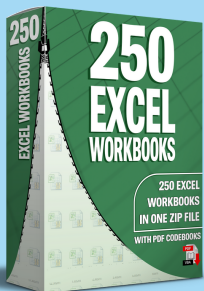
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



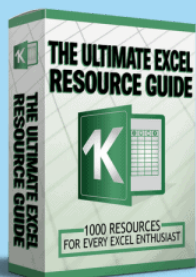
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



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

[Click here to learn more](#)



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

[Click here to learn more](#)

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Worksheet_Change [Sub].....	2
Worksheet_SelectionChange [Sub].....	2
Dashboard.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub].....	3
Entries.....	4
(Declarations).....	4
Worksheet_Change [Sub].....	4
Worksheet_SelectionChange [Sub].....	4
EntryDB.....	6
(Declarations).....	6
Pivots.....	7
(Declarations).....	7
Sheet1.....	8
(Declarations).....	8
ThisWorkbook.....	9
(Declarations).....	9
Modules.....	10
Admin_Macros.....	10
(Declarations).....	10
Account_ListRefresh [Sub].....	10
AccountAndEntry_ListRefresh [Sub].....	10
EntryType_ListRefresh [Sub].....	10
GoToSheet [Sub].....	11
Dashboard_Macros.....	12
(Declarations).....	12
Dashboard_Expense [Sub].....	12
Dashboard_Income [Sub].....	12
Dashboard_NetWorth [Sub].....	12
Entry_Macros.....	13
(Declarations).....	13
Entry_CreateMultipleEntries [Sub].....	13
Entry_Delete [Sub].....	13
Entry_Load [Sub].....	14
Entry_LoadEntries [Sub].....	14
Entry_LoadEntryTypes [Sub].....	14
Entry_New [Sub].....	15
Entry_SavedMsg [Sub].....	15
Entry_SaveUpdate [Sub].....	15
PopUp_Calendar.....	16
(Declarations).....	16
CalCol [Sub].....	16
CalendarHide [Sub].....	16
CalendarShow [Sub].....	16
CalFormulaReplacement [Sub].....	17
CheckForSheet [Sub].....	17
CreateCalSht [Sub].....	18
GroupCal [Sub].....	20
MacroLinkRemover [Sub].....	20
NextMonth [Sub].....	21
NextYear [Sub].....	21
PrevMonth [Sub].....	21
PrevYear [Sub].....	22
ReplaceCalendar [Sub].....	22
SelectDay [Sub].....	22
ShowSettings [Sub].....	22
UnGroupCal [Sub].....	23

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("E8:F34" )) Is Nothing Then EntryType_ListRefresh
5     If Not Intersect(Target, Range("G8:I34" )) Is Nothing Then Account_ListRefresh
6 End Sub
7
8 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
9
10 End Sub
```

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5     If Not Intersect(Target, Range("D2:F2" )) Is Nothing Then
6         If Not Intersect(Target, Range("D2" )) Is Nothing Then Dashboard_Income
7         If Not Intersect(Target, Range("E2" )) Is Nothing Then Dashboard_Expense
8         If Not Intersect(Target, Range("F2" )) Is Nothing Then Dashboard_NetWorth
9         Range("D6" ).Select
10    End If
11 End Sub
12
```

```
1 Option Explicit
```

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

```
4
5     If Not Intersect(Target, Range("N5" )) Is Nothing And Range("N5" ).Value <> Empty
```

```
6     Then
```

```
7         Dim EntryType As String
```

```
8         EntryType = Range("N5" ).Value 'Set Entry type
```

```
9         Range("N9" ).Validation.Delete 'Delete validation first
```

```
10        Select Case EntryType
```

```
11            Case Is = "Income"
```

```
12                Range("N9" ).Validation.Add xlValidateList, AlertStyle:=xlValidAlertStop,
```

```
13                Formula1:="=Income_Type"
```

```
14            Case Is = "Expense"
```

```
15                Range("N9" ).Validation.Add xlValidateList, AlertStyle:=xlValidAlertStop,
```

```
16                Formula1:="=Expense_Type"
```

```
17            Case Is = "Payment" , "Transfer"
```

```
18                Range("N9" ).Validation.Add xlValidateList, AlertStyle:=xlValidAlertStop,
```

```
19                Formula1:="=AllAcctTypes"
```

```
20        End Select
```

```
21    End If
```

```
22    'On Show Entry Name Change, Run Macro to populate list
```

```
23    If Not Intersect(Target, Range("F4" )) Is Nothing Then Entry_LoadEntryTypes
```

```
24    'On Entry Date Change, refresh entry list
```

```
25    If Not Intersect(Target, Range("I4,K4" )) Is Nothing Then Entry_LoadEntries
```

```
26 End Sub
```

```
27 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
28     If Target.CountLarge > 2 Then Exit Sub
```

```
29     'On Entry Type Change, Update Name List
```

```
30     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
```

```
31     it in
```

```
32     Dim Cal As Shape
```

```
33     On Error Resume Next
```

```
34     Set Cal = Shapes("Calendar" )
```

```
35     On Error GoTo 0
```

```
36     If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
```

```
37     If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
```

```
38     If Not Intersect(Target, Range("I4,K4,P7" )) Is Nothing Then 'Date Range Selection
```

```
39         CheckForSheet
```

```
40         CalendarShow
```

```
41     Else:
```

```
42         CheckForSheet
```

```
43         CalendarHide
```

```
44     End If
```

```
45     'On Entry Type Selection
```

```
46     If Not Intersect(Target, Range("E6:F9999" )) Is Nothing And Range("E" & Target.Row)
```

```
47     .Value <> "" Then
```

```
48         EntryDB.Range("T3" ).Value = Range("F" & Target.Row).Value 'Set Entry Name
```

```
49         Range("B6" ).Value = Target.Row 'Set Selected Row
```

```
50         Entry_LoadEntries
```

```
51     End If
```

```
52     'On Entry Selection
```

```
53     If Not Intersect(Target, Range("H6:K9999" )) Is Nothing And Range("G" & Target.Row)
```

```
1 2  
54     .Value <> "" Then  
55         Range("B3").Value = Range("G" & Target.Row).Value 'Set Entry ID  
56         Range("B7").Value = Target.Row 'Set Selected Row  
56         Entry_Load  
57     End If  
58 End Sub
```


1 Option Explicit

2

1 Option Explicit

2

1 Option Explicit

2

1 Option Explicit

2

```

1 Option Explicit
2 Dim LastRow As Long, ListRow As Long, LastIncRow As Long, LastExpRow As Long,
  LastAssetRow As Long, LastLiabRow As Long
3
4 Sub Account_ListRefresh()
5     With Admin
6         .Range("B8:B999" ).ClearContents 'Clear existing values
7         LastAssetRow = .Range("G99999" ).End(xlUp).Row 'Last Asset Row
8         LastLiabRow = .Range("I99999" ).End(xlUp).Row 'Last Liability Row
9         ListRow = LastAssetRow + 1 'Set First Exp Row
10        .Range("B8:B" & LastAssetRow).Value = .Range("G8:G" & LastAssetRow).Value
        'Copy Over Asset Accounts
11        .Range("B" & ListRow & ":B" & ListRow + LastLiabRow - 8).Value = .Range("I8:I"
        & LastLiabRow).Value 'Copy Over Liabilities
12        LastRow = .Range("B99999" ).End(xlUp).Row
13        'Sort Income & Expenses
14        If LastRow < 9 Then Exit Sub
15        With .Sort
16            .SortFields.Clear
17            .SortFields.Add Key:=Admin.Range("B8" ), SortOn:=xlSortOnValues, Order:=
        xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
18            .SetRange Admin.Range("B8:B" & LastRow)
19            .Apply
20        End With
21    End With
22 End Sub
23
24 Sub AccountAndEntry_ListRefresh()
25     Dim LastEntryRow As Long, LastAcctRow As Long
26     With Admin
27         .Range("AA8:AA9999" ).ClearContents
28         LastEntryRow = .Range("A99999" ).End(xlUp).Row
29         LastAcctRow = .Range("B99999" ).End(xlUp).Row
30         .Range("AA8:AA" & LastEntryRow).Value = .Range("A8:A" & LastEntryRow).Value
        'Copy Over entry Items
31         .Range("AA" & LastEntryRow + 1 & ":AA" & LastEntryRow + LastAcctRow - 7).Value
        = .Range("B8:B" & LastAcctRow).Value
32         LastRow = .Range("AA99999" ).End(xlUp).Row
33         'Sort Accounts & Entries
34         If LastRow < 9 Then Exit Sub
35         With .Sort
36             .SortFields.Clear
37             .SortFields.Add Key:=Admin.Range("AA8" ), SortOn:=xlSortOnValues, Order:=
        xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
38             .SetRange Admin.Range("AA8:AA" & LastRow)
39             .Apply
40         End With
41    End With
42 End Sub
43
44 Sub EntryType_ListRefresh()
45     With Admin
46         .Range("A8:A999" ).ClearContents 'Clear existing values
47         LastIncRow = .Range("E99999" ).End(xlUp).Row 'Last income Row
48         LastExpRow = .Range("F99999" ).End(xlUp).Row 'Last Expense Row
49         ListRow = LastIncRow + 1 'Set First Exp Row
        .Range("A8:A" & LastIncRow).Value = .Range("E8:E" & LastIncRow).Value 'Copy
        Over Income Types
50        .Range("A" & ListRow & ":A" & ListRow + LastExpRow - 8).Value = .Range("F8:F"
        & LastExpRow).Value 'Copy Over Expenses
51        LastRow = .Range("A99999" ).End(xlUp).Row

```

1 2

```
1 2
52 'Sort Income & Expenses
53 If LastRow < 9 Then Exit Sub
54 With .Sort
55     .SortFields.Clear
56     .SortFields.Add Key:=Admin.Range("A8" ), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
57     .SetRange Admin.Range("A8:A" & LastRow)
58     .Apply
59 End With
60 End With
61 End Sub
62 Sub GoToSheet()
63     ThisWorkbook.Sheets(Application.Caller).Activate
64 End Sub
```

```
1 Option Explicit
2
3 Sub Dashboard_Expense()
4     With Dashboard
5         .Range("B2").Value = 5 'Set Selected column
6         .Shapes("IncomeGrp").Visible = msoFalse
7         .Shapes("ExpenseGrp").Visible = msoCTrue
8         .Shapes("NetWorthGrp").Visible = msoFalse
9     End With
10 End Sub
11
12
13 Sub Dashboard_Income()
14     With Dashboard
15         .Range("B2").Value = 4 'Set Selected column
16         .Shapes("IncomeGrp").Visible = msoCTrue
17         .Shapes("ExpenseGrp").Visible = msoFalse
18         .Shapes("NetWorthGrp").Visible = msoFalse
19     End With
20 End Sub
21
22 Sub Dashboard_NetWorth()
23     With Dashboard
24         .Range("B2").Value = 6 'Set Selected column
25         .Shapes("IncomeGrp").Visible = msoFalse
26         .Shapes("ExpenseGrp").Visible = msoFalse
27         .Shapes("NetWorthGrp").Visible = msoCTrue
28     End With
29 End Sub
```

```

1 Option Explicit
2 Dim EntryRow As Long, EntryCol As Long, LastRow As Long, LastResultRow As Long
3 Dim EntryID As String
4
5 Sub Entry_CreateMultipleEntries()
6     Dim EntryFreq As String
7     Dim FreqQty As Long, EntryQty As Long, EntryNumb As Long
8     Dim EntryDt As Date, StartDt As Date, NextDt As Date
9     Dim InfAmnt As Double, InfRate As Double, EntryAmnt As Double
10
11     With Entries
12         If .Range("B9" ).Value < 4 Then
13             MsgBox "Please make sure to fill in all 4 Multiple Entry fields in order to
14                 create multiple entries"
15             Exit Sub
16         End If
17
18         If .Range("B8" ).Value < 5 Then
19             MsgBox "Please make sure to add an Entry Type, Entry Date, Income/Expense &
20                 Account before saving this Entry"
21             Exit Sub
22         End If
23
24         FreqQty = .Range("O16" ).Value 'Frequency Qty
25         EntryFreq = .Range("P16" ).Value 'Frequency
26         StartDt = .Range("N17" ).Value 'Starting Date
27         EntryQty = .Range("P17" ).Value 'Entry Qty
28         NextDt = StartDt 'Set Intial Next Date
29         InfRate = .Range("P18" ).Value 'Annual Inflation Rate
30         .Range("B11" ).Value = True 'Set Multiple Entries to True
31         For EntryNumb = 1 To EntryQty
32             EntryAmnt = .Range("P11" ).Value 'Entry Amount
33             .Range("P7" ).Value = NextDt 'Set Entry Date
34             .Range("B3" ).ClearContents 'Clear Assigned ID
35             .Range("P11" ).Value = EntryAmnt + InfAmnt 'Account for inflation (if any)
36             Entry_SaveUpdate
37             Select Case EntryFreq
38                 Case Is = "Week(s)"
39                     NextDt = DateAdd("ww" , FreqQty, NextDt)
40                     If InfRate <> 0 Then InfAmnt = EntryAmnt * (InfRate / 52)
41                 Case Is = "Month(s)"
42                     NextDt = DateAdd("m" , FreqQty, NextDt)
43                     If InfRate <> 0 Then InfAmnt = EntryAmnt * (InfRate / 12)
44                 Case Is = "Year(s)"
45                     NextDt = DateAdd("yyyy" , FreqQty, NextDt)
46                     If InfRate <> 0 Then InfAmnt = EntryAmnt * InfRate
47             End Select
48         Next EntryNumb
49         MsgBox EntryQty & " " & .Range("N9" ).Value & " entries have been added"
50         .Range("B11" ).Value = False 'Set Create Multiple to False
51     End With
52 End Sub
53
54
55
56
57
58 Sub Entry_Delete()
59     1

```



```

1
59  If MsgBox("Are you sure you want to delete this Entry?" , vbYesNo, "Delete Entry" )
60  = vbNo Then Exit Sub
61  If Entries.Range("B4" ).Value = Empty Then GoTo NotSaved
62  EntryRow = Entries.Range("B4" ).Value 'Set Entry Row
63  EntryDB.Range(EntryRow & ":" & EntryRow).EntireRow.Delete
64  NotSaved:
65  Entry_New 'Clear Fields
66  Entry_LoadEntries 'Reload Entry List
67  End Sub
68
68  Sub Entry_Load()
69  With Entries
70  .Range("N7,P7:Q7,N9,P9:Q9,N11,P11:Q11,N13:Q13" ).ClearContents
71  If .Range("B4" ).Value = Empty Then
72  MsgBox "Please select a correct Entry to load"
73  Exit Sub
74  End If
75  EntryRow = .Range("B4" ).Value 'Entry Row
76  For EntryCol = 2 To 10
77  .Range(EntryDB.Cells(1, EntryCol).Value).Value = EntryDB.Cells(EntryRow,
78  EntryCol).Value 'Load Data
79  Next EntryCol
80  End With
81  End Sub
82
82  Sub Entry_LoadEntries()
83  Entries.Range("B7,G6:K999" ).ClearContents
84  With EntryDB
85  .Range("X3:AB9999" ).ClearContents 'Clear Prior Results
86  LastRow = .Range("A99999" ).End(xlUp).Row
87  If LastRow < 4 Then Exit Sub
88  .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
89  "T2:V3" ), CopyToRange:=.Range("X2:AA2" ), Unique:=True
90  LastResultRow = .Range("X99999" ).End(xlUp).Row
91  If LastResultRow < 3 Then Exit Sub
92  .Range("AB3:AB" & LastResultRow).Formula = .Range("AB1" ).Formula 'Copy down
93  difference formula
94  Entries.Range("G6:K" & LastResultRow + 3).Value = .Range("X3:AB" &
95  LastResultRow).Value 'Bring Over Entry Details
96  End With
97  With Entries
98  If .Range("B3" ).Value <> Empty Then
99  On Error Resume Next
100 .Range("B7" ).Value = .Range("G6:G999" ).Find(.Range("B3" ).Value, ,
101 xlFormulas, xlWhole).Row 'Set Selected Entry Type Row (if any)
102 On Error GoTo 0
103 End If
104 End With
105 End Sub
106
106  Sub Entry_LoadEntryTypes()
107  Entries.Range("B6,B7,E6:F999" ).ClearContents 'Clear selected Rows & Data
108  With EntryDB
109  LastRow = .Range("A99999" ).End(xlUp).Row
110  If LastRow < 4 Then Exit Sub
111  .Range("A3:F" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
112  "O2:O3" ), CopyToRange:=.Range("Q2:R2" ), Unique:=True
113  LastResultRow = .Range("Q99999" ).End(xlUp).Row
114  If LastResultRow < 3 Then Exit Sub
115  Entries.Range("E6:F" & LastResultRow + 3).Value = .Range("Q3:R" & LastResultRow
116  ).Value 'Bring Over Entry Types

```

1 2

```

111     End With
112     With Entries
113         If .Range("N9" ).Value <> Empty Then
114             On Error Resume Next
115             .Range("B6" ).Value = .Range("F6:F999" ).Find(.Range("N9" ).Value, , xlValues,
116                 xlWhole).Row 'Set Selected Entry Type Row (if any)
117         End If
118     End With
119 End Sub
120 Sub Entry_New()
121     Entries.Range("B3,N7,N9,P7:Q7,N9,P9:Q9,N11,P11:Q11,N13:Q13" ).ClearContents
122     Entries.Range("P7" ).Value = Date 'Set Intial Entry date
123     Entries.Range("N7" ).Select 'Select intial cell
124 End Sub
125 Sub Entry_SavedMsg()
126     With Entries.Shapes("EntrySavedMsg" )
127         Dim i As Long, Delay As Double, StartTime As Double
128         .Visible = msoCTrue
129         For i = 1 To 100
130             .Fill.Transparency = i / 100
131             Delay = 0.009
132             StartTime = Timer
133             Do
134                 DoEvents
135             Loop While Timer - StartTime < Delay
136         Next i
137         .Visible = msoFalse
138     End With
139 End Sub
140 Sub Entry_SaveUpdate()
141     With Entries
142         If .Range("B8" ).Value < 5 Then
143             MsgBox "Please make sure to add an Entry Type, Entry Date, Income/Expense &
144                 Account before saving this Entry"
145             Exit Sub
146         End If
147         'Check For New or Existing Entry
148         If .Range("B4" ).Value = "" Then 'New Entry
149             EntryRow = EntryDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail. Entry
150             Row
151             .Range("B3" ).Value = .Range("B5" ).Value 'Set Entry ID
152             EntryDB.Range("A" & EntryRow).Value = .Range("B3" ).Value 'Entry ID
153         Else 'Existing Entry
154             EntryRow = .Range("B4" ).Value 'Existing Entry Row
155         End If
156         For EntryCol = 2 To 10
157             EntryDB.Cells(EntryRow, EntryCol).Value = .Range(EntryDB.Cells(1, EntryCol).
158                 Value).Value 'Save/Update Data
159         Next EntryCol
160         Entry_LoadEntryTypes ''Reload Entry Types
161         Entry_LoadEntries 'Reload Entry List
162         If .Range("B11" ).Value = False Then Entry_SavedMsg 'Run saved Message, except
163             when creating multiple entries
164     End With
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 = Selection
37         'Check if active cell is a valid date
38         If IsDate(SelCell.Value) = True Then
39             Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
40         Else: 'If No Date or incorrect Date user current date
41             Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
42         End If
43         'Clear all shapes to white (if calendar is visible)
44         If ActiveSheet.Shapes("Calendar" ).Visible = True Then
45             For DayNum = 1 To 42
46                 DayName = DayNum & "Day"
47
48                 With ActiveSheet.Shapes(DayName)
49                     .Fill.ForeColor.RGB = RGB(255, 255, 255)
50                     .TextFrame2.TextRange.Font.Bold = msoFalse
51                 End With
52             Next DayNum
53         End If
54
55         Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).
           Value) 'Set Month
56         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
           Value) 'Set Year
57         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"

```

1 2

```

1 2
58 ' UnGroupCal
59 If InStr(.Shapes("Calendar").GroupItems("NextTri").OnAction, "!") <> 0 Or
   InStr(.Shapes("1Day").DrawingObject.Formula, "]") <> 0 Then 'Run Workbook
   Link Remover and Cell Link Replacement
60     MacroLinkRemover
61     CalFormulaReplacement
62 End If
63 'GroupCal
64 On Error GoTo NoCal
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").Top = SelCell.Offset(1, 0).Top
75 .Shapes("Calendar").ZOrder msoBringToFront
76 If Sheets("CalPopUp").Range("A6").Value > 0 Then
77     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
   "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

```

```

1
111 Set ws = ActiveWorkbook.Sheets("CalPopUp" )
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"

```

1 2

```

1 2
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 =
"=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

```

```

1
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" ,
"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
1

```

```

1
250 For Each Shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
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)

```

1 2 3


```

1 2 3
307     .Fill.ForeColor.RGB = RGB(255, 255, 255)
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     CalPopUp.Shapes("Calendar" ).Copy 'Copy From Developers sheet
335     ActiveCell.Select 'Select the active Cell
336     ActiveSheet.Paste 'Paste in Calendar
337 End Sub
338
339 .....
340
341 ''''Select Day Of The Month
342 .....
343
344 Sub SelectDay()
345     Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
346     DayNumb = Replace(Application.Caller, "Day" , "" )
347     RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
348     ColNumb = DayNumb Mod 7 + 1
349     If DayNumb Mod 7 = 0 Then
350         ColNumb = 8
351     Else
352         ColNumb = DayNumb Mod 7 + 1
353     End If
354
355     'On Error Resume Next
356     If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
357     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
358         ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
359     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
360     ActiveCell.Offset(0, 1).Select
361 End Sub
362
363 Sub ShowSettings()
364     'Show or Hide Calendar Settings Panel
365     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
366         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
367             "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
368             = False
369     Else:
370         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,

```

```
1 2  
    "CalCol14" , "CalCol15" , "CalCol16" , "CalCol17" , "CalCol18" , "CalCol19" ))).Visible  
    = True  
361 End If  
362 End Sub  
363  
364 Sub UngroupCal()  
365     On Error Resume Next  
366     ActiveSheet.Shapes("Calendar" ).Ungroup  
367     ActiveSheet.Shapes("NextMonth" ).Ungroup  
368     ActiveSheet.Shapes("PrevMonth" ).Ungroup  
369     On Error GoTo 0  
370 End Sub
```

—, 20

A

Account_ListRefresh, 2, 10
 AccountAndEntry_ListRefresh, 10
 Activate, 11, 18
 ActiveCell, 17, 22
 ActiveSheet, 16-18, 20-23
 ActiveWorkbook, 18, 19
 ActSht, 18
 Add, 4, 10, 11, 18, 19
 Address, 17, 18
 Admin, 10, 11
 AdvancedFilter, 14
 After, 18
 AlertStyle, 4
 Application, 11, 16, 22
 Apply, 10, 11
 Array, 16, 17, 20-22
 AutoFill, 19

B

Bold, 16, 17, 21, 22

C

Cal, 4
 CalCol, 16, 18
 CalendarHide, 4, 16
 CalendarShow, 4, 16
 CalFormulaReplacement, 17
 Caller, 11, 16, 22
 CalPopUp, 22
 Cells, 14, 15, 17, 18, 22
 CheckForSheet, 4, 17
 Chr, 18, 19
 Clear, 10, 11
 ClearContents, 10, 13-15
 ColCnt, 18
 ColNum, 17
 ColNumb, 22
 Copy, 18, 22
 CopyToRange, 14
 Count, 18
 CountLarge, 3, 4
 CreateCalSht, 18
 CreateWS, 17, 18
 CriteriaRange, 14

D

Dashboard, 12
 Dashboard_Expense, 3, 12
 Dashboard_Income, 3, 12
 Dashboard_NetWorth, 3, 12
 DataOption, 10, 11
 DateAdd, 13
 DayCnt, 18
 DayName, 16, 17, 21
 DayNum, 16, 17
 DayNumb, 22
 Delay, 15
 Delete, 4, 14
 Destination, 19
 DoEvents, 15
 DrawingObject, 17, 18

E

Empty, 4, 14-16, 21, 22
 EntireRow, 14

Entries, 13-15
 Entry_CreateMultipleEntries, 13
 Entry_Delete, 13
 Entry_Load, 5, 14
 Entry_LoadEntries, 4, 14, 15
 Entry_LoadEntryTypes, 4, 14, 15
 Entry_New, 14, 15
 Entry_SavedMsg, 15
 Entry_SaveUpdate, 13, 15
 EntryAmnt, 13
 EntryCol, 13-15
 EntryDB, 4, 14, 15
 EntryDt, 13
 EntryFreq, 13
 EntryID, 13
 EntryNumb, 13
 EntryQty, 13
 EntryRow, 13-15
 EntryType, 4
 EntryType_ListRefresh, 2, 10
 Explicit, 2-4, 6-10, 12, 13

F

Fill, 15-17, 21, 22
 Find, 14, 15
 Font, 16, 17, 21, 22
 ForeColor, 16, 17, 21, 22
 Formula, 14, 17, 18
 Formula1, 4
 FreqQty, 13

G

GoToSheet, 11
 Group, 20
 GroupCal, 19, 20
 GroupItems, 17

I

i, 15
 InfAmnt, 13
 InfRate, 13
 InStr, 17, 21
 Intersect, 2-4
 IsDate, 16

K

Key, 10, 11

L

LastAcctRow, 10
 LastAssetRow, 10
 LastEntryRow, 10
 LastExpRow, 10
 LastIncRow, 10
 LastLiabRow, 10
 LastResultRow, 13, 14
 LastRow, 10, 11, 13, 14
 Left, 17, 21
 Len, 21
 ListRow, 10

M

MacroLink, 20, 21
 MacroLinkRemover, 17, 20
 Month, 16, 18
 MsgBox, 13-15, 17, 20
 msoBringToFront, 17
 msoCTrue, 12, 15, 17, 20
 msoFalse, 12, 15, 16, 21, 22
 msoTrue, 17

N

Name, 18-20
 Names, 19
 NewLink, 20, 21
 NextDt, 13
 NextMonth, 21
 NextShp, 21
 NextYear, 21
 NoCal, 16, 17, 20
 NotSaved, 14
 NumberFormat, 19

O

Offset, 17, 22
 OnAction, 17, 18, 21
 Order, 10, 11

P

Paste, 18, 22
 Placement, 17, 20
 PrevMonth, 21
 PrevYear, 22

R

Range, 2-5, 10-22
 RefersTo, 19
 Replace, 22
 ReplaceCalendar, 4, 22
 RGB, 16, 17, 21, 22
 Right, 21
 RoundUp, 22
 Row, 4, 5, 10, 14, 15
 RowCnt, 18
 RowNum, 17
 RowNumb, 22

S

SelCell, 16, 17
 SelectDay, 22
 Selection, 16, 20
 SetRange, 10, 11
 Shape, 4, 17, 20
 ShapeRange, 16, 20
 Shapes, 4, 12, 15-18, 20-23
 Sheets, 11, 16-18, 21, 22
 ShowSettings, 22
 Shp, 17, 20, 21
 Sort, 10, 11
 SortFields, 10, 11
 SortOn, 10, 11
 Split, 21
 SplitLink, 20, 21
 StartDt, 13
 StartTime, 15

T

Target, 2-5
 TextFrame2, 16, 17, 21, 22
 TextRange, 16, 17, 21, 22
 ThisWorkbook, 11, 18, 21, 22
 Timer, 15
 Top, 17
 Transparency, 15

U

Ungroup, 23
 UnGroupCal, 18, 23
 Unique, 14

V

Validation, 4
Value, 4, 5, 10, 12-19, 21, 22
vbNo, 14
vbYesNo, 14
Visible, 4, 12, 15-18, 20-23

W

Worksheet, 17, 18
Worksheet_Change, 2, 4
Worksheet_SelectionChange, 2-4
WorksheetFunction, 22
ws, 17, 18

X

xlAscending, 10, 11
xlFillDefault, 19
xlFilterCopy, 14
xlFormulas, 14
xlMove, 17, 20
xlSheetHidden, 18
xlSortNormal, 10, 11
xlSortOnValues, 10, 11
xlUp, 10, 14, 15
xlValidAlertStop, 4
xlValidateList, 4
xlValues, 15
xlWhole, 14, 15

Y

Year, 16, 18

Z

ZOrder, 17

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