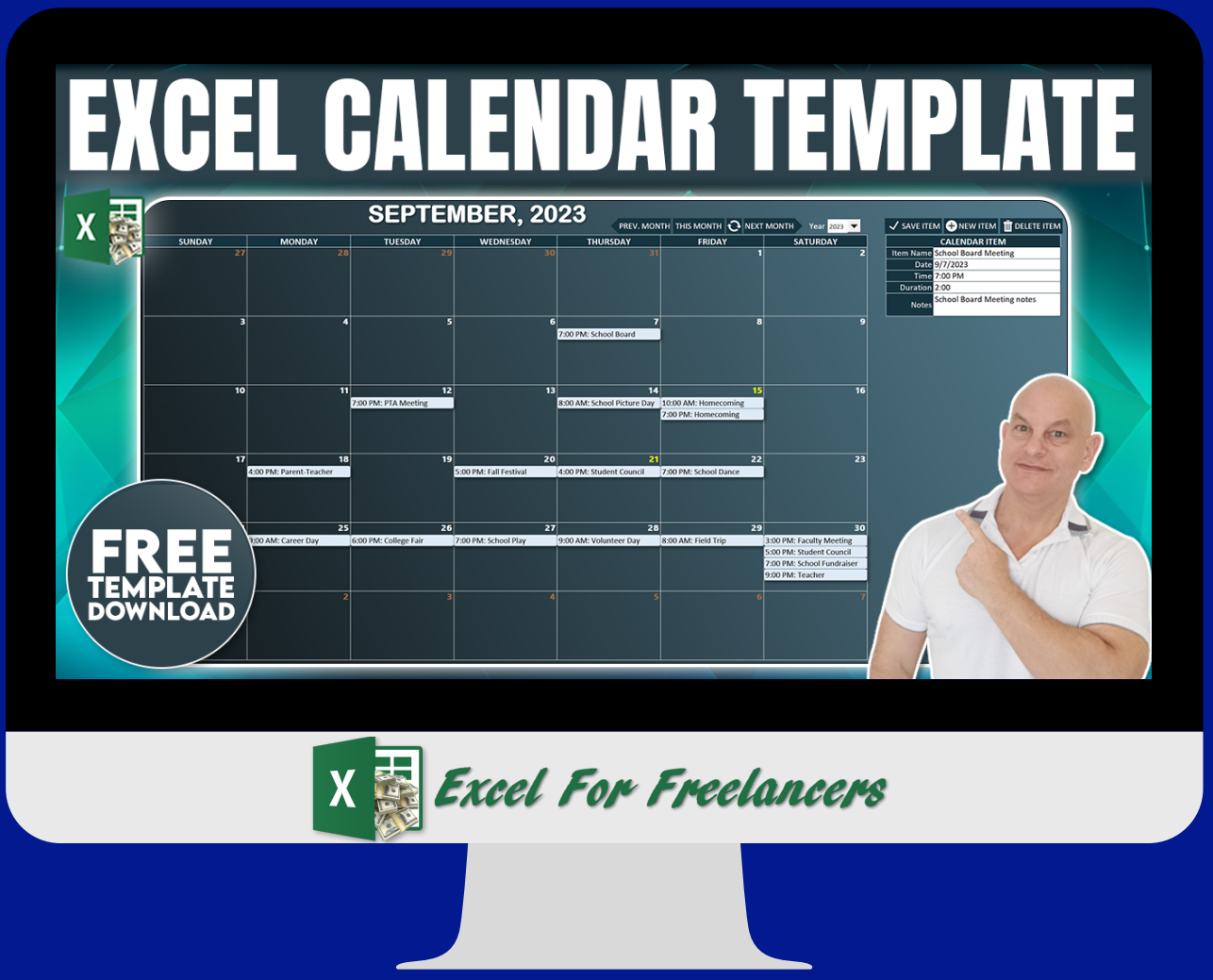
VBA SOURCE CODE BOOK



Learn How To Create This 2023 Calendar Template In Excel

DOWNLOAD APPLICATION

VIEW TRAINING

by: Randy Austin

ABOUT THE AUTHOR

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

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

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

WEBSITE

YOUTUBE

FACEBOOK

TWITTER

DISCORD

INSTAGRAM

TELEGRAM

RUMBLE

Microsoft® MVP Most Valuable Professional

OUR COURSES & PRODUCTS

MENTORSHIP EXPRESS

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

Click here to learn more

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

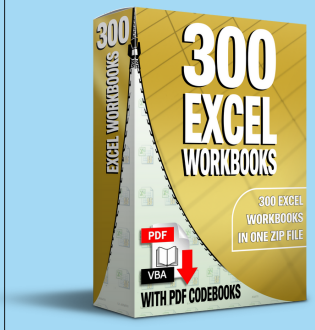
Click here to learn more

-

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

Click here to learn more

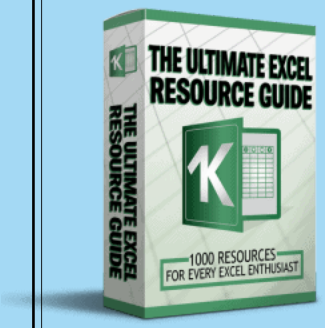
-



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

Click here to learn more

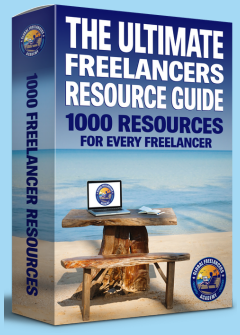
OUR COURSES & PRODUCTS



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

Click here to learn more

-



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

Click here to learn more

-

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

Click here to learn more

Table of Contents

```
Option Explicit

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
  If Target.CountLarge > 1 Then Exit Sub
  If Shapes("ColorPalette" ).Visible = True Then Shapes("ColorPalette" ).Visible =
  msoFalse

  'Show color palette
  If Not Intersect(Target, Range("F7,F9" )) Is Nothing Then
    With Shapes("ColorPalette" )
      .Left = Range("F"  & Target.Row).Left
      .Top = Range("F"  & Target.Row + 1).Top
      .Visible = msoCTrue
    End With
  End If
End Sub
```

```
1   Option Explicit
2
```

```
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
```

```
Option Explicit
Dim CalRow As Long, CalCol As Long, LastRow As Long, LastResultRow As Long, ResultRow As Long
Dim ItemNumb As Long, DayItemCount As Long, ItemTop
Dim ItemColor As String, ItemID As String, ItemName As String
Dim ItemShp As Shape
Dim ItemDate As Date, ItemTime As String
Dim ItemWidth As Double, ItemLeft As Double


Sub Calendar_CheckForMove()
  Dim DestRow As Long, DestCol As Long, ItemRow As Long, ItemCol As Long, CountDelay As Long
  With Calendar
    If .Range("B10" ).Value = Empty Then Exit Sub
    ItemID = .Range("B9" ).Value 'Item ID

    For CountDelay = 1 To 100000
      DoEvents
      If .Range("B1" ).Value = True Then End 'Exit Loop on Move Item Move = True

      With .Shapes("CalItem"  & ItemID)
        If .Left <> Calendar.Range("B15" ).Value Or .Top <> Calendar.Range("B16" ).Value Then 'Move Detect
          'Check for incorrect move
          If .Left < Calendar.Range("D1" ).Left Or .Left > Calendar.Range("K1" ).Left Or .Top > Calendar.Range("A39" ).Top Or .Top < Calendar.Range("A4" ).Top - 1 Then
            MsgBox "Plase make sure to move the Calendar Item to a correct calendar date on the Calendar"
            Calendar_Refresh
            Exit Sub
          End If

          DestRow = Calendar.Shapes("CalItem"  & ItemID).TopLeftCell.Row
          DestCol = Calendar.Shapes("CalItem"  & ItemID).TopLeftCell.Column
          DestRow = DestRow - (DestRow + 2) Mod 6 'Date Row
          ItemDate = Calendar.Cells(DestRow, DestCol).Value   'Calendar Date
          If ItemDate = 0 Then
            MsgBox "Please move the item to a day with an existing Date"
            Calendar_Refresh
            Exit Sub
          End If
          Calendar.Range("M5" ).Value = ItemDate 'new Item dates
          Item_SaveUpdate 'Save Item Changes
          Calendar.Range("B1" ).Value = True 'Set Move Item to true
          End
        End If
      End With
    Next CountDelay
    .Range("B1" ).Value = True  'Set Move Item Movement to True
  End With
End Sub


Sub Calendar_Item_Select()
  With Calendar
    ItemID = Replace(Application.Caller, Left(Application.Caller, 7), "" ) 'Extract Item ID
```

```
    For Each ItemShp In .Shapes 'Reset All Colors back to default color
      If InStr(ItemShp.Name, "CalItem" ) > 0 Then ItemShp.Fill.ForeColor.RGB = Admin.Range("F7" ).Interior.Color
    Next ItemShp
    .Shapes("CalItem"  & ItemID).Fill.ForeColor.RGB = Admin.Range("F9" ).Interior.Color
    .Range("B9" ).Value = ItemID
    'Drag & Drop Settings
    Calendar.Range("B8" ).Value = ItemID
    .Range("B15" ).Value = Calendar.Shapes(Application.Caller).Left 'set Selected Left Position
    .Range("B16" ).Value = Calendar.Shapes(Application.Caller).Top ' Set Selected Top Position
    .Range("B1" ).Value = False 'Set Calendar Move to False
    .Shapes("CalItem"  & ItemID).Select
    Item_Load
    Calendar_CheckForMove
  End With
End Sub

Sub Calendar_NextMonth()
  With Calendar
    If .Range("B2" ).Value = 7 And .Range("B5" ).Value = 12 Then
      MsgBox "You are at the last year and first last available"
      Exit Sub
    End If
    If .Range("B5" ).Value = 12 Then 'December Month
      .Range("B2" ).Value = .Range("B2" ).Value + 1 'Increase Year by 1
      .Range("B5" ).Value = 1 'Set To January
    Else 'Not January
      .Range("B5" ).Value = .Range("B5" ).Value + 1 'Increase month by 1
    End If
    Calendar_Refresh 'Refresh Calendar
  End With
End Sub

Sub Calendar_PrevMonth()
  With Calendar
    If .Range("B2" ).Value = 1 And .Range("B5" ).Value = 1 Then
      MsgBox "You are at the first year and first month available"
      Exit Sub
    End If
    If .Range("B5" ).Value = 1 Then 'January Month
      .Range("B2" ).Value = .Range("B2" ).Value - 1 'Reduce Year by 1
      .Range("B5" ).Value = 12 'Set To December
    Else 'Not January
      .Range("B5" ).Value = .Range("B5" ).Value - 1 'Reduce month by 1
    End If
    Calendar_Refresh 'Refresh Calendar
  End With
End Sub

Sub Calendar_Refresh()
  'Clear all existing item shapes from Calendar sheet
  For Each ItemShp In Calendar.Shapes
    If InStr(ItemShp.Name, "CalItem" ) > 0 Then ItemShp.Delete
  Next ItemShp
  ItemColor = Admin.Range("F7" ).Interior.Color 'Set Item Shape Color
  ItemNumb = 1 'Set default item # to 1
  ItemWidth = 1 'Set Default to 1
```

```
110     With ItemsDB
111       LastRow = .Range("A99999" ).End(xlUp).Row 'Last Item Row
112       If LastRow < 4 Then Exit Sub
113       Application.ScreenUpdating = False
114       .Range("A3:F"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
          "I2:J3" ), CopyToRange:=.Range("M2:Q2" ), Unique:=True
115       LastResultRow = .Range("M99999" ).End(xlUp).Row
116       If LastResultRow < 3 Then Exit Sub
117       If LastResultRow < 4 Then GoTo NoSort
118       With .Sort
119         .SortFields.Clear
120         .SortFields.Add Key:=ItemsDB.Range("O3" ), SortOn:=xlSortOnValues, Order:=
          xlAscending, DataOption:=xlSortNormal  'Sort Based On Date
121         .SortFields.Add Key:=ItemsDB.Range("P3" ), SortOn:=xlSortOnValues, Order:=
          xlAscending, DataOption:=xlSortNormal  'Sort Based On Time
122         .SetRange ItemsDB.Range("M3:Q"  & LastResultRow) 'Set Range
123         .Apply 'Apply Sort
124       End With
125       NoSort:
126       For ResultRow = 3 To LastResultRow
127         ItemID = .Range("M"  & ResultRow).Value 'item ID
128         ItemName = .Range("N"  & ResultRow).Value 'Item Name
129         ItemDate = .Range("O"  & ResultRow).Value 'Item Date
130         ItemTime = Format(.Range("P"  & ResultRow).Value, "h:mma/p" )
131         DayItemCount = Application.WorksheetFunction.CountIf([ItemDate_Results],
          ItemDate) 'Get # of Items in a single Date
132         If DayItemCount > 5 Then
133           If DayItemCount > 10 Then DayItemCount = 10 'Set Item Limit to 10
134           ItemWidth = 0.5 'Set Item with to 1/2
135         End If
136         For CalRow = 4 To 34 Step 6
137           For CalCol = 4 To 10
138             If Calendar.Cells(CalRow, CalCol).Value = ItemDate Then 'Day Found
139               Calendar.Shapes("SampleItemShp" ).Duplicate.Name = "CalItem"  &
                  ItemID
140               With Calendar.Shapes("CalItem"  & ItemID)
141                 .Left = Calendar.Cells(CalRow, CalCol).Left + ItemLeft 'Set Left
                    Pos.
142                 .Top = Calendar.Cells(CalRow + ItemNumb + ItemTop, CalCol).Top + 1

143                 .Width = Calendar.Cells(CalRow + ItemNumb, CalCol).Width *
                    ItemWidth
144                 .Height = Calendar.Cells(CalRow + ItemNumb, CalCol).Height
145                 .TextFrame2.TextRange.Text = ItemTime & ": "  & ItemName 'Text
                    inside shape
146                 .Fill.ForeColor.RGB = ItemColor 'Set Item Color
147                 .OnAction = "Calendar_Item_Select"  'Macro to select item
148               End With
149               If ItemNumb >= DayItemCount Then
150                 ItemNumb = 1 'Reset
151                 ItemWidth = 1 'Reset Item Width
152               Else
153                 ItemNumb = ItemNumb + 1 'Increment by 1
154               End If
155               If ItemNumb <= 5 Then
156                 ItemLeft = 0 'Set Left Position
157                 ItemTop = 0
158               Else 'Item Number from 6 to 10
159                 ItemLeft = Calendar.Cells(CalRow + ItemNumb, CalCol).Width / 2
160                 ItemTop = -5
```

```
1 2 3 4 5 6 7
161             End If
162           End If
163         Next CalCol
164       Next CalRow
165     Next ResultRow
166   End With
167   Application.ScreenUpdating = True
168 End Sub
169
170 Sub Calendar_ThisMonth()
171   With Calendar
172     .Range("B5" ).Value = Month(Date) 'Set Current Month #
173     .Range("B2" ).Value = Admin.Range("Years" ).Find(Year(Date), , xlFormulas,
        xlWhole).Row - 3
174     Calendar_Refresh 'Refresh Calendar
175   End With
176 End Sub
177
178 Sub PrintCalendar()
179   Calendar.PrintOut , , , False, True, , , , False
180 End Sub
```

```
1   Option Explicit
2
3   Dim ItemRow As Long, ItemCol As Long
4
5   Sub Item_Delete()
6     If MsgBox("Are you sure you want to delete this Calendar Item?" , vbYesNo, "Delete
      Item" ) = vbNo Then Exit Sub
7     With Calendar
8       If .Range("B10" ).Value = Empty Then GoTo NotSaved
9       ItemRow = .Range("B10" ).Value 'Item row
10      ItemsDB.Range(ItemRow & ":"  & ItemRow).EntireRow.Delete 'Delete ROw
11      NotSaved:
12      Item_New
13      Calendar_Refresh 'Refresh calendar
14    End With
15  End Sub
16
17  Sub Item_Load()
18    With Calendar
19      If .Range("B10" ).Value = Empty Then
20        MsgBox "Please select a correct calendar item to load"
21        Exit Sub
22      End If
23      ItemRow = .Range("B10" ).Value 'Item Row
24      For ItemCol = 2 To 6
25        .Range("M"  & ItemCol + 2).Value = ItemsDB.Cells(ItemRow, ItemCol).Value
          'Bring Over data
26      Next ItemCol
27    End With
28  End Sub
29
30  Sub Item_New()
31    Calendar.Range("B9,M4:M9" ).ClearContents 'Clear Out
32  End Sub
33
34  Sub Item_SaveUpdate()
35    With Calendar
36      If Application.WorksheetFunction.CountA(.Range("M4:M6" )) < 3 Then
37        MsgBox "Please make sure to add in an Item Name, Date & Time"
38        Exit Sub
39      End If
40      If .Range("B10" ).Value = Empty Then 'New Item
41        ItemRow = ItemsDB.Range("A99999" ).End(xlUp).Row + 1  'First Avail. Row
42        .Range("B9" ).Value = .Range("B11" ).Value 'Set Item ID
43        ItemsDB.Range("A"  & ItemRow).Value = .Range("B9" ).Value 'Item ID
44      Else 'Existing Item
45        ItemRow = .Range("B10" ).Value 'Existing Item Row
46      End If
47      For ItemCol = 2 To 6
48        ItemsDB.Cells(ItemRow, ItemCol).Value = .Range("M"  & ItemCol + 2).Value
          'Save/ Update data
49      Next ItemCol
50      If .Range("B12" ).Value = False Then 'Refresh & Set Saved Message except for
        recurring
51        Calendar_Refresh 'Refresh calendar
52        MsgBox "Item Saved"
53      End If
54    End With
55  End Sub
```

```
Option Explicit

Sub CreateRecurringItem()
  Dim ItemDate As Date
  Dim Freq As String, FreqQty As Long, ItemQty As Long, ItemNumb As Long
  With Calendar
    If .Range("B10" ).Value = Empty Then
      MsgBox "Please make sure to save this item before creating recurring calendar
      item"
      Exit Sub
    End If
    If Application.WorksheetFunction.CountA(.Range("M12:M15" )) < 4 Then
      MsgBox "Please make sure to fill in all recurring fields"
      Exit Sub
    End If
    FreqQty = .Range("M12" ).Value 'Frequency Qty
    Freq = .Range("M13" ).Value 'Frequency
    ItemDate = .Range("M14" ).Value  'Start Date
    ItemQty = .Range("M15" ).Value ' Number of Calendar Items
    .Range("B12" ).Value = True 'Set Recurring to True
    For ItemNumb = 1 To ItemQty
      .Range("M5" ).Value = ItemDate 'Cal. Item Date
      .Range("B9" ).ClearContents 'Clear Existing Item ID
      Item_SaveUpdate 'Save Calendar Item
      Select Case Freq
      Case Is = "Minute(s)"
        ItemDate = DateAdd("n" , FreqQty, ItemDate)
      Case Is = "Hour(s)"
        ItemDate = DateAdd("h" , FreqQty, ItemDate)
      Case Is = "Day(s)"
        ItemDate = DateAdd("d" , FreqQty, ItemDate)
      Case Is = "Week(s)"
        ItemDate = DateAdd("ww" , FreqQty, ItemDate)
      Case Is = "Month(s)"
        ItemDate = DateAdd("m" , FreqQty, Now)
      End Select
    Next ItemNumb
    .Range("B12" ).Value = False 'Set Recurring to false
    Calendar_Refresh 'Refresh calendar
    MsgBox ItemQty & " calendar items have been created"
  End With
End Sub
```

```
1   Option Explicit
2
3   Sub SendAppointmentToOutlook()
4     Dim ItemDate As Date, ItemTime As Date
5     Dim ItemName As String, Notes As String
6     Dim Dur As Double
7     With Calendar
8       ItemName = .Range("M4" ).Value 'item Name
9       ItemDate = .Range("M5" ).Value 'Item Date
10      ItemTime = .Range("M6" ).Value 'Time
11      Dur = .Range("M7" ).Value * 24 * 60 'Duration (in minutes)
12      Notes = .Range("M8" ).Value 'Notes
13    End With
14
15
16    Dim OutApp As Object, OutAppt As Object
17    Set OutApp = CreateObject("Outlook.Application" )
18    Set OutAppt = OutApp.CreateItem(1)
19    With OutAppt
20      .Subject = ItemName
21      .Start = ItemDate + ItemTime
22      .Duration = Dur
23      .ReminderSet = True
24      .ReminderMinutesBeforeStart = 15
25      .Body = Notes
26      .Save
27      .Display
28    End With
29    Set OutApp = Nothing
30    Set OutAppt = Nothing
31    MsgBox "Calendar Item has been sent to Outlook"
32  End Sub
```

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