

VBA SOURCE CODE BOOK

CONSTRUCTION ESTIMATOR - UPDATED



Excel For Freelancers

How To Create A Construction Estimator Manager



[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 400,000 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



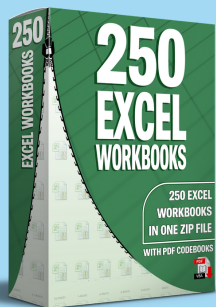
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[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
Clients.....	3
(Declarations).....	3
Items.....	4
(Declarations).....	4
ProjDB.....	5
(Declarations).....	5
Projects.....	6
(Declarations).....	6
Worksheet_Change [Sub].....	6
Worksheet_SelectionChange [Sub].....	7
ProjItemDB.....	8
(Declarations).....	8
ThisWorkbook.....	9
(Declarations).....	9
Forms.....	10
AddEditClient.....	10
(Declarations).....	10
CancelBtn_Click [Sub].....	10
SaveBtn_Click [Sub].....	10
AddEditItem.....	11
(Declarations).....	11
CancelBtn_Click [Sub].....	11
SaveBtn_Click [Sub].....	11
Modules.....	12
Client_Macros.....	12
(Declarations).....	12
Client_AddEdit [Sub].....	12
Client_BuildClientList [Sub].....	12
Client_SaveUpdate [Sub].....	12
Item_Macros.....	14
(Declarations).....	14
Item_AddEdit [Sub].....	14
Item_SaveUpdate [Sub].....	14
Project_Macros.....	16
(Declarations).....	16
Project_CretePDF [Sub].....	16
Project_Delete [Sub].....	16
Project_Email [Sub].....	16
Project_Generate [Sub].....	17
Project_Load [Sub].....	17
Project_New [Sub].....	18
Project_Print [Sub].....	18
Project_SavedMsg [Sub].....	18
Project_SaveUpdate [Sub].....	19
ProjSheet_Macros.....	21
(Declarations).....	21
Category_TabUpdates [Sub].....	21
CategoryTab_Select [Sub].....	21
Proj_NextProj [Sub].....	22
Project_PrevProj [Sub].....	22

```
1 Option Explicit
2
3
4 Private Sub Worksheet_Change(ByVal Target As Range)
5     If Not Intersect(Target, Range("C7:C14" )) Is Nothing Then Category_TabUpdates
6
7     'On Date Column Default Change
8     If Not Intersect(Target, Range("J11" )) Is Nothing Then
9         If Target.Value = "Yes" Then Projects.Range("R:R" ).EntireColumn.Hidden = False
10        Else Projects.Range("R:R" ).EntireColumn.Hidden = True
11    End If
12
13    'On Cost Column Default Change
14    If Not Intersect(Target, Range("J12" )) Is Nothing Then
15        If Target.Value = "Yes" Then Projects.Range("W:X" ).EntireColumn.Hidden = False
16        Else Projects.Range("W:X" ).EntireColumn.Hidden = True
17    End If
18 End Sub
```

1
2

Option Explicit

1 Option Explicit

2

1
2

Option Explicit


```
1 Option Explicit
```

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

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

```
5 If Not Intersect(Target, Range("N2" )) Is Nothing And Range("N2" ).Value <> Empty
6 And Range("B2" ).Value <> Empty Then Project_Load
```

```
7 'On Client Change, Load Address & Client ID
```

```
8 If Not Intersect(Target, Range("F2" )) Is Nothing And Range("F2" ).Value <> Empty
9 Then
```

```
10 Dim ClientRow As Long
```

```
11 If Range("B8" ).Value = Empty Then
```

```
12 Range("F3:F4" ).ClearContents 'Clear Fields on New Client
```

```
13 If MsgBox(Range("F2" ).Value & " is not currently a client. Would you like to
14 add this client?" , vbYesNo, "New Client" ) = vbYes Then Client_AddEdit
```

```
15 Exit Sub
```

```
16 End If
```

```
17 ClientRow = Range("B8" ).Value 'Client Row
```

```
18 Range("B7" ).Value = Clients.Range("A" & ClientRow).Value 'Client ID
```

```
19 Range("F3" ).Value = Clients.Range("C" & ClientRow).Value 'Address
```

```
20 Range("F4" ).Value = Clients.Range("D" & ClientRow).Value & ", " & Clients.
```

```
21 Range("E" & ClientRow).Value & " " & Clients.Range("F" & ClientRow).Value
```

```
22 End If
```

```
23 'On Item change but not on project/item load
```

```
24 If Not Intersect(Target, Range("E9:E200" )) Is Nothing And Range("B1" ).Value =
25 False Then
```

```
26 Dim ItemRow As Long, ProjItemDBRow As Long
```

```
27 If Target.Value <> Empty Then
```

```
28 On Error Resume Next
```

```
29 ItemRow = Items.Range("Item_Name" ).Find(Target.Value, , xlValues, xlWhole).
30 Row
```

```
31 On Error GoTo 0
```

```
32 If ItemRow = 0 Then
```

```
33 If MsgBox(Target.Value & " is not currently an item. Would you like to add
34 this item?" , vbYesNo, "New Item" ) = vbYes Then Item_AddEdit
```

```
35 Exit Sub
```

```
36 End If
```

```
37 Range("F" & Target.Row).Value = Items.Range("D" & ItemRow).Value 'Set
38 Description
```

```
39 Range("G" & Target.Row).Value = 1 'Intial Qty to 1
```

```
40 Range("H" & Target.Row).Value = Items.Range("E" & ItemRow).Value 'Unit type
```

```
41 Range("I" & Target.Row).Value = Items.Range("F" & ItemRow).Value 'Item Cost
```

```
42 Range("J" & Target.Row).Formula = "=G" & Target.Row & "*I" & Target.Row
```

```
43 'Total Cost Formula
```

```
44 Range("K" & Target.Row).Value = Items.Range("G" & ItemRow).Value 'Sales
```

```
45 Price
```

```
46 Range("L" & Target.Row).Formula = "=G" & Target.Row & "*K" & Target.Row
```

```
47 'Total Price Formula
```

```
48 Range("M" & Target.Row).Formula = Items.Range("A" & ItemRow).Value 'Item ID
```

```
49 Else 'Empty (clear row and database if applicable)
```

```
50 If Range("N" & Target.Row).Value <> Empty Then 'Remove from DB
```

```
51 ProjItemDBRow = Range("N" & Target.Row).Value 'Project Item DB Row
```

```
52 ProjItemDB.Range(ProjItemDBRow & ":" & ProjItemDBRow).EntireRow.Delete
```

```
53 End If
```

```
54 Range("F" & Target.Row & ":" & Target.Row).ClearContents 'Clear Entire row
```

```
55 End If
```

```
1 2
```

```
47     1 2 End If
48 End Sub
49
50 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
51     If Target.CountLarge > 1 Then Exit Sub
52     If Shapes("AddEditItemBtn").Visible = True Then Shapes("AddEditItemBtn").Visible =
53         msoFalse 'Hide Add/Edit item Btn
54     If Not Intersect(Target, Range("D9:L299")) Is Nothing Then
55         Range("B4").Value = Target.Row 'Set Selected row
56         Range("B10").Value = Range("M" & Target.Row).Value 'Add Item ID (if any)
57     If Not Intersect(Target, Range("E9:E200")) Is Nothing Then
58         With Shapes("AddEditItemBtn")
59             .Left = Range("F" & Target.Row).Left + 15
60             .Top = Range("F" & Target.Row).Top
61             .Visible = msoCTrue
62         End With
63     End If
64 End Sub
```

1
2

Option Explicit

1
2

Option Explicit

```
1  
2  
3 Option Explicit  
4  
5 Private Sub CancelBtn_Click()  
6     AddEditClient.Hide  
7 End Sub  
8  
9  
10 Private Sub SaveBtn_Click()  
11     Client_SaveUpdate  
12 End Sub
```

```
1  
2  
3 Option Explicit  
4  
5 Private Sub CancelBtn_Click()  
6     AddEditItem.Hide  
7 End Sub  
8  
9  
10 Private Sub SaveBtn_Click()  
11     Item_SaveUpdate  
12 End Sub
```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long
3 Dim ClientFld As Control
4 Dim ClientRow As Long, ClientCol As Long
5
6 Sub Client_AddEdit()
7     Unload AddEditClient 'Clear Fields In Form
8     With Projects
9         If .Range("B8" ).Value = Empty Then 'New Client
10            AddEditClient.Field1.Value = .Range("F2" ).Value
11        Else 'Existing Client
12            ClientRow = .Range("B8" ).Value 'Set Client Row
13            For ClientCol = 1 To 7
14                Set ClientFld = AddEditClient.Controls("Field" & ClientCol)
15                ClientFld.Value = Clients.Cells(ClientRow, ClientCol + 1).Value
16            Next ClientCol
17        End If
18        AddEditClient.Show 'Display Form
19    End With
20 End Sub
21 Sub Client_BuildClientList()
22     With Clients
23         LastRow = .Range("A99999" ).End(xlUp).Row
24         If LastRow < 3 Then Exit Sub
25         .Range("B3:B" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
26             "K2" ), Unique:=True
27         LastResultRow = .Range("K99999" ).End(xlUp).Row
28         If LastResultRow < 4 Then Exit Sub 'Exit on 1 or no results
29         'Sort Alphabetically
30         With .Sort
31             .SortFields.Clear
32             .SortFields.Add Key:=Clients.Range("K3" ), SortOn:=xlSortOnValues, Order:=
33                 xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
34             .SetRange Clients.Range("K3:K" & LastResultRow)
35             .Apply
36         End With
37     End With
38 End Sub
39 Sub Client_SaveUpdate()
40     If AddEditClient.Field1.Value = Empty Then
41         MsgBox "Please make sure to add a Client Name before saving"
42         Exit Sub
43     End If
44     With Clients
45         'Check For new or Existing Client
46         If Projects.Range("B8" ).Value = Empty Then 'New Client
47             ClientRow = .Range("A99999" ).End(xlUp).Row + 1 'First avail Row
48             Projects.Range("B7" ).Value = Projects.Range("B9" ).Value 'Set Client ID
49             .Range("A" & ClientRow).Value = Projects.Range("B9" ).Value 'Unique Client
50             ID
51         Else 'Existing Client
52             ClientRow = Projects.Range("B8" ).Value 'Existing Client Row
53         End If
54         For ClientCol = 2 To 8
55             Set ClientFld = AddEditClient.Controls("Field" & ClientCol - 1)
56             .Cells(ClientRow, ClientCol).Value = ClientFld.Value 'Adds/Updates Field Date
57             in db
58         Next ClientCol
59         AddEditClient.Hide
60         Client_BuildClientList 'Run Macro to update & sort Clients

```

1 2

```
58 | 1 2 Projects.Range("F2").Value = AddEditClient.Field1.Value ' Set Client on  
    |   | Projects  
59 |   | End With  
60 |   | End Sub
```



```

1 Option Explicit
2 Dim SelRow As Long, ItemRow As Long, ItemCol As Long, LastRow As Long
3 Dim ItemFld As Control
4
5
6 Sub Item_AddEdit()
7     Unload AddEditItem 'Clear Fields In Form
8     With Projects
9         SelRow = .Range("B4").Value
10        If SelRow = 0 Then Exit Sub
11
12        If .Range("B11").Value = Empty Then 'New Item
13            AddEditItem.Field1.Value = .Range("E" & SelRow).Value 'Entered Item Name
14            AddEditItem.ItemCat.Value = .Range("B5").Value 'Set Default Category
15        Else 'Existing Item
16            ItemRow = .Range("B11").Value 'Set Item Row
17            AddEditItem.ItemCat.Value = Items.Range("B" & ItemRow).Value 'Set Item
18            Categor
19            For ItemCol = 3 To 7
20                Set ItemFld = AddEditItem.Controls("Field" & ItemCol - 2)
21                ItemFld.Value = Items.Cells(ItemRow, ItemCol).Value
22            Next ItemCol
23        End If
24        AddEditItem.Show 'Display Form
25    End With
26 End Sub
27
28 Sub Item_SaveUpdate()
29     If AddEditItem.Field1.Value = Empty Then
30         MsgBox "Please make sure to add an Item Name before saving"
31         Exit Sub
32     End If
33     SelRow = Projects.Range("B4").Value 'Set Selected Row
34     With Items
35         'Check For new or Existing Item
36         If Projects.Range("B11").Value = Empty Then 'New Item
37             ItemRow = .Range("A99999").End(xlUp).Row + 1 'First avail Row
38             Projects.Range("B10").Value = Projects.Range("B12").Value 'Set Item ID
39             .Range("A" & ItemRow).Value = Projects.Range("B10").Value 'Unique Item ID
40         Else 'Existing Item
41             ItemRow = Projects.Range("B11").Value 'Existing Item Row
42         End If
43         Items.Range("B" & ItemRow).Value = AddEditItem.ItemCat.Value 'Set Item Category
44
45         For ItemCol = 3 To 7
46             Set ItemFld = AddEditItem.Controls("Field" & ItemCol - 2)
47             .Cells(ItemRow, ItemCol).Value = ItemFld.Value 'Adds/Updates Field Date in db
48         Next ItemCol
49         AddEditItem.Hide
50         'Reset Selected Item List
51         With Items
52             LastRow = .Range("A99999").End(xlUp).Row
53             If LastRow < 4 Then Exit Sub
54             .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
55                 "J2:J3"), CopyToRange:=.Range("L2"), Unique:=True
56         End With
57         Projects.Range("E" & SelRow).Value = AddEditItem.Field1.Value ' Set Item on
58         Projects
59     End With
60 End Sub

```

57

```

1 Option Explicit
2 Dim ProjRow As Long, ProjCol As Long, LastItemRow As Long, ItemRow As Long, ResultRow
  As Long, LastResultRow As Long, ProjDBRow As Long, CatRow As Long
3 Dim TypeNum As Long, ProjNum As Long, ProjID As Long, ProjectItemRow As Long,
  ProjItemDBRow As Long, LastProjItemDBRow As Long, IDRow As Long
4 Dim SelCat As String, FileName As String
5 Dim OutApp As Object, OutMail As Object
6 Sub Project_CretePDF()
7     Project_Generate 'Generate the complete project for printing
8     FileName = ThisWorkbook.Path & "\" & Projects.Range("F2" ).Value & "_Project_" &
  Projects.Range("N2" ).Value & ".pdf"
9     If Dir(FileName, vbDirectory) <> "" Then Kill (FileName) 'Delete file if exists
10    Projects.ExportAsFixedFormat xlTypePDF, FileName, , , False, , , True
11 End Sub
12
13 Sub Project_Delete()
14     If MsgBox("Are you sure you want to delete this Project?" , vbYesNo, "Delete
  Project" ) = vbNo Then Exit Sub
15     If Projects.Range("B2" ).Value = "" Then GoTo NotSaved 'Project Not yet saved
16     ProjRow = Projects.Range("B2" ).Value 'Project Row
17     ProjDB.Range(ProjRow & ":" & ProjRow).EntireRow.Delete 'Delete Project DB Row
18     With ProjItemDB
19         LastProjItemDBRow = .Range("A99999" ).End(xlUp).Row
20         .Range("A3:N" & LastProjItemDBRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.
  Range("Q2:Q3" ), CopyToRange:=.Range("U2:AE2" ), Unique:=True
21         LastResultRow = .Range("AE99999" ).End(xlUp).Row 'Last Results Row
22         If LastResultRow < 3 Then GoTo NoDBItems
23         If LastResultRow < 4 Then GoTo NoSort
24         With .Sort
25             .SortFields.Clear
26             .SortFields.Add Key:=ProjItemDB.Range("AE3" ), Order:=xlDescending
27             .SetRange ProjItemDB.Range("U3:AE" & LastResultRow)
28             .Apply
29         End With
30         NoSort:
31         For ResultRow = 3 To LastResultRow
32             ProjItemDBRow = .Range("AE" & ResultRow).Value 'Project Item Row
33             .Range(ProjItemDBRow & ":" & ProjItemDBRow).EntireRow.Delete
34         Next ResultRow
35         NoDBItems:
36         NotSaved:
37         MsgBox "This project has been deleted"
38         Project_New
39     End With
40 End Sub
41
42
43
44 Sub Project_Email()
45     Project_Generate 'Generate the complete project for printing
46     With Projects
47         If .Range("B2" ).Value = Empty Then
48             MsgBox "Please make sure to save this project before emailing"
49         End If
50
51         FileName = ThisWorkbook.Path & "\" & .Range("F2" ).Value & "_Project_" & .Range
  ("N2" ).Value & ".pdf"
52         If Dir(FileName, vbDirectory) <> "" Then Kill (FileName) 'Delete file if exists
53
54         .ExportAsFixedFormat xlTypePDF, FileName, , , False
55         Set OutApp = CreateObject("Outlook.Application" )

```

```

1 2
55     Set OutMail = OutApp.createItem(0)
56
57     With OutMail
58         .To = Clients.Range("H" & Projects.Range("B8" ).Value).Value 'Client Email
59         .attachments.Add FileName
60         .Subject = "Please see the attached project_" & Projects.Range("L1" ).Value
61         'Set Default Subject Line
62         .body = ""
63         .display 'To send use .Send
64     End With
65     On Error GoTo 0
66     Set OutMail = Nothing
67 End With
68 End Sub
69
70 Sub Project_Generate()
71     Projects.Range("R7:Z9999" ).ClearContents
72     ItemRow = 7 'Set Intial Item Row
73     With ProjItemDB
74         LastProjItemDBRow = ProjItemDB.Range("A9999" ).End(xlUp).Row 'Last Item Row
75         If LastProjItemDBRow < 4 Then Exit Sub
76         Application.ScreenUpdating = False
77         For CatRow = 7 To 14
78             .Range("U4:AF9999" ).ClearContents 'Clear Previous Data
79             If Admin.Range("C" & CatRow).Value = Empty Then GoTo NextCat
80             .Range("P3" ).Value = Admin.Range("C" & CatRow).Value 'Set Category
81             Criteria
82             ProjItemDB.Range("A3:N" & LastProjItemDBRow).AdvancedFilter xlFilterCopy,
83             CriteriaRange:=ProjItemDB.Range("P2:Q3" ), CopyToRange:=ProjItemDB.Range(
84             "U2:AF2" ), Unique:=True
85             LastResultRow = ProjItemDB.Range("V9999" ).End(xlUp).Row 'Last Results Row
86             If LastResultRow < 3 Then GoTo NoCatItems
87             Projects.Range("S" & ItemRow).Value = UCase(Admin.Range("C" & CatRow).Value)
88             'Set Category (in uppercse)
89             ItemRow = ItemRow + 1
90             Projects.Range("R" & ItemRow & ":Z" & ItemRow + LastResultRow - 3).Value = .
91             Range("U3:AC" & LastResultRow).Value 'Bring Over Category Items
92             ItemRow = ItemRow + LastResultRow - 2 'Update Next Item Row
93             NextCat:
94             NoCatItems:
95         Next CatRow
96     End With
97     'Set Footer
98     With Projects.Shapes("FooterGrp" )
99         .Left = Projects.Range("R1" ).Left
100        .Top = Projects.Range("R" & ItemRow).Top
101        .Width = Projects.Range("R:Z" ).Width 'Set With
102        .Visible = msoCTrue
103    End With
104    'Set Print Area
105    Projects.PageSetup.PrintArea = "$R$1:$Z$" & ItemRow + 2
106    Application.ScreenUpdating = True
107 End Sub
108
109 Sub Project_Load()
110     With Projects
111         If .Range("B2" ).Value = Empty Then
112             MsgBox "Please enter a correct Project #"
113         Exit Sub
114     End With
115 End Sub

```

```

1 2 3
110 End If
111 ProjRow = .Range("B2" ).Value 'Project Row
112 .Range("B1" ).Value = True 'Set Project load to true
113 .Range("B4,B7,B10,F2:F4,I2:J4,N3,N4,D9:N200" ).ClearContents 'Clear Project
Fields
114 Application.ScreenUpdating = False
115 For ProjCol = 2 To 9 'Do not include Totals (Cost, Sales Price, Profit)
116 .Range(ProjDB.Cells(1, ProjCol).Value).Value = ProjDB.Cells(ProjRow, ProjCol).
Value 'Bring in Project data
117 Next ProjCol
118
119 'Load Project Items
120 LastProjItemDBRow = ProjItemDB.Range("A99999" ).End(xlUp).Row 'Last Item Row
121 If LastProjItemDBRow < 4 Then GoTo NoDBItems
122 ProjItemDB.Range("A3:N" & LastProjItemDBRow).AdvancedFilter xlFilterCopy,
CriteriaRange:=ProjItemDB.Range("Q2:R3" ), CopyToRange:=ProjItemDB.Range(
"U2:AF2" ), Unique:=True
123 LastResultRow = ProjItemDB.Range("V99999" ).End(xlUp).Row 'Last Results Row
124 If LastResultRow < 3 Then GoTo NoDBItems
125 For ResultRow = 3 To LastResultRow
126 ItemRow = ProjItemDB.Range("AF" & ResultRow).Value 'Project Item Row
127 .Range("D" & ItemRow & ":N" & ItemRow).Value = ProjItemDB.Range("U" &
ResultRow & ":AE" & ResultRow).Value 'Item Name, Desc. Qty, Amounts
128 .Range("J" & ItemRow).Formula = "=G" & ItemRow & "*I" & ItemRow 'Total
Cost Formula
129 .Range("L" & ItemRow).Formula = "=G" & ItemRow & "*K" & ItemRow 'Total
Price Formula
130 Next ResultRow
131 NoDBItems:
132 .Range("B1" ).Value = False 'Set Project load to False
133 Application.ScreenUpdating = True
134 End With
135 End Sub
136
137 Sub Project_New()
138 With Projects
139 .Range("B1" ).Value = True 'Set Project load to true
140 .Range("B4,B7,B10,F2:F4,I2:J4,N2:N4,D9:N200" ).ClearContents 'Clear Project
Fields
141 .Range("N2" ).Value = .Range("B3" ).Value 'Set Next Project ID
142 .Range("N3" ).Value = Date 'Set current date
143 .Range("N4" ).Value = Admin.Range("J13" ).Value 'Default Status
144 .Range("O6" ).Value = Admin.Range("J14" ).Value 'Deafult Footer message
145 .Range("B1" ).Value = False 'Set Project load to false
146 .Shapes("AddEditItemBtn" ).Visible = msoFalse 'Hide Add/Edit Item Button
147 .Range("F2" ).Select 'Select Client Cell
148 End With
149 End Sub
150
151 Sub Project_Print()
152 Project_Generate 'Generate the complete project for printing
153 Projects.PrintOut , , , , True, , , , False
154 End Sub
155
156 Sub Project_SavedMsg()
157 With Projects.Shapes("ProjSavedMsg" )
158 Dim i As Long, Delay As Double, StartTime As Double
159 .Visible = msoCTrue
160 For i = 1 To 100
161 .Fill.Transparency = i / 100
162 Delay = 0.009

```

```

1 2 3
163     StartTime = Timer
164     Do
165         DoEvents
166     Loop While Timer - StartTime < Delay
167     Next i
168     .Visible = msoFalse
169 End With
170 End Sub
171 Sub Project_SaveUpdate()
172     With Projects
173         'Check For Required Fields
174         If .Range("B8" ).Value = Empty Then 'Missing Client
175             MsgBox "Please be sure to add a client to this project"
176             .Range("F2" ).Select
177             Exit Sub
178         End If
179         If .Range("N3" ).Value = Empty Then 'Missing date
180             MsgBox "Please be sure to add an Project Date before saving this project"
181             .Range("N3" ).Select
182             Exit Sub
183         End If
184         If .Range("B2" ).Value = Empty Then 'New Project
185             ProjRow = ProjDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail. row
186             .Range("N2" ).Value = .Range("B3" ).Value 'Next Project Id
187             ProjDB.Range("A" & ProjRow).Value = .Range("B3" ).Value 'Project ID
188         Else 'Existing Project
189             ProjRow = .Range("B2" ).Value 'existing Project Row
190         End If
191         ProjID = .Range("N2" ).Value 'Project ID
192         SelCat = .Range("B5" ).Value 'Selected Category
193         For ProjCol = 2 To 12
194             ProjDB.Cells(ProjRow, ProjCol).Value = .Range(ProjDB.Cells(1, ProjCol).Value).
195             Value 'Save Data to Projects database
196         Next ProjCol
197         'Add/Update Project Items
198         LastItemRow = .Range("E200" ).End(xlUp).Row 'Last Item Row
199         If LastItemRow < 9 Then GoTo NoItems
200         For ItemRow = 9 To LastItemRow
201             If .Range("N" & ItemRow).Value <> Empty Then 'Existing DB Row
202                 ProjItemDBRow = .Range("N" & ItemRow).Value 'Project item Row
203             Else 'New DB Row
204                 ProjItemDBRow = ProjItemDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail
205
206                 ProjItemDB.Range("A" & ProjItemDBRow).Value = ProjID 'Project ID
207                 ProjItemDB.Range("B" & ProjItemDBRow).Value = SelCat 'Selected Category
208                 ProjItemDB.Range("M" & ProjItemDBRow).Value = ItemRow 'Project Item Row
209                 ProjItemDB.Range("N" & ProjItemDBRow).Value = "=Row()" 'Database Row
210                 .Range("N" & ItemRow).Value = ProjItemDBRow 'Set Database Row
211             End If
212             ProjItemDB.Range("C" & ProjItemDBRow).Value = .Range("M" & ItemRow).Value
213             'Item ID
214             ProjItemDB.Range("D" & ProjItemDBRow & ":L" & ProjItemDBRow).Value = .Range(
215             "D" & ItemRow & ":L" & ItemRow).Value 'All Item Details
216         Next ItemRow
217         NoItems:
218         On Error Resume Next
219         If Application.Caller = "SaveBtn" Then Project_SavedMsg 'Run fade out message
220         (only on manual save)
221         On Error GoTo 0
222     End With

```

219

1

End Sub

```

1 Option Explicit
2 Dim CatQty As Long, CatNumb As Long, SelCat As Long, LastRow As Long, ProjRow As Long
3 Dim ItemRow As Long, LastResultRow As Long, ResultRow As Long, LastProjItemDBRow As
  Long
4 Dim CatWidth As Double, LeftPos As Double, TopPos As Double
5 Sub Category_TabUpdates()
6   CatQty = Application.WorksheetFunction.CountA(Admin.Range("Categories" ))
7
8   With Projects
9     CatWidth = .Range("D:L" ).Width / CatQty
10    LeftPos = .Range("D7" ).Left 'Set Intial Left Position
11    TopPos = .Range("D7" ).Top + 2 'Set Top Position
12    For CatNumb = 1 To 8
13      If CatNumb <= CatQty Then
14        With .Shapes("Category" & CatNumb)
15          .Left = LeftPos
16          .Top = TopPos
17          .Width = CatWidth
18          .Visible = msoCTrue
19        End With
20        LeftPos = LeftPos + CatWidth 'Increase Left Position
21      Else 'Hidden/Non existing categories
22        .Shapes("Category" & CatNumb).Visible = msoFalse
23      End If
24    Next CatNumb
25  End With
26 End Sub
27
28
29 Sub CategoryTab_Select()
30   With Projects
31     If .Range("B2" ).Value = Empty Then
32       MsgBox "Plase save this project before switching tabs"
33       Exit Sub
34     End If
35     Application.ScreenUpdating = False
36     Project_SaveUpdate 'Save before switching tabs
37     SelCat = Replace(Application.Caller, "Category" , "" )
38     .Range("B6" ).Value = SelCat 'Category #
39     For CatNumb = 1 To 8 'Reset All tabs to unselected color and not bold
40       .Shapes("Category" & CatNumb).Fill.TwoColorGradient msoGradientHorizontal, 2
41       .Shapes("Category" & CatNumb).TextFrame2.TextRange.Font.Bold = msoFalse
42     Next CatNumb
43     .Shapes("Category" & SelCat).Fill.TwoColorGradient msoGradientHorizontal, 1
44     .Shapes("Category" & SelCat).TextFrame2.TextRange.Font.Bold = msoCTrue 'Set
45     Bold On Selected Tab
46   End With
47   'Set Selected Item List
48   With Items
49     LastRow = .Range("A99999" ).End(xlUp).Row
50     If LastRow < 4 Then Exit Sub
51     .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
52       "J2:J3" ), CopyToRange:=.Range("L2" ), Unique:=True
53   End With
54   'Add Items From Selected Tab
55   Projects.Range("B1" ).Value = True
56   Projects.Range("B4,D9:N200" ).ClearContents 'Clear Existing Items
57   Projects.Shapes("AddEditItemBtn" ).Visible = msoFalse 'Hide Add/Edit Item Button

```

1


```

1
58 With ProjItemDB
59 LastProjItemDBRow = .Range("A99999").End(xlUp).Row 'Last Item Row
60 If LastProjItemDBRow < 4 Then GoTo NoDBItems
61 .Range("A3:N" & LastProjItemDBRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.
Range("Q2:R3"), CopyToRange:=.Range("U2:AF2"), Unique:=True
62 LastResultRow = .Range("V99999").End(xlUp).Row 'Last Results Row
63 If LastResultRow < 3 Then GoTo NoDBItems
64 For ResultRow = 3 To LastResultRow
65 ItemRow = .Range("AF" & ResultRow).Value 'Project Item Row
66 Projects.Range("D" & ItemRow & ":N" & ItemRow).Value = .Range("U" &
ResultRow & ":AE" & ResultRow).Value 'Item Name, Desc. Qty, Amounts
67 Projects.Range("J" & ItemRow).Formula = "=G" & ItemRow & "*I" & ItemRow
'Total Cost Formula
68 Projects.Range("L" & ItemRow).Formula = "=G" & ItemRow & "*K" & ItemRow
'Total Price Formula
69 Next ResultRow
70 NoDBItems:
71 Projects.Range("B1").Value = False 'Set Project load to False
72 Application.ScreenUpdating = True
73 End With
74 End Sub

75
76 Sub Proj_NextProj()
77 With Projects
78 If .Range("N2").Value = Application.WorksheetFunction.Max(ProjDB.Range(
"Proj_ID")) Then
79 MsgBox "You are at the last project"
80 Exit Sub
81 End If
82 On Error Resume Next
83 ProjRow = ProjDB.Range("Proj_ID").Find(.Range("N2").Value, , xlValues, xlWhole)
.Row
84 On Error GoTo 0
85 If ProjRow <> 0 Then
86 .Range("N2").Value = ProjDB.Range("A" & ProjRow + 1).Value 'Set Next
Project ID
87 Project_Load
88 End If
89 End With
90 End Sub

91
92
93 Sub Project_PrevProj()
94 With Projects
95 If .Range("N2").Value = Application.WorksheetFunction.Min(ProjDB.Range(
"Proj_ID")) Then
96 MsgBox "You are at the first project"
97 Exit Sub
98 End If
99 On Error Resume Next
100 ProjRow = ProjDB.Range("Proj_ID").Find(.Range("N2").Value, , xlValues, xlWhole)
.Row
101 On Error GoTo 0
102 If ProjRow <> 0 And ProjRow <> 4 Then
103 .Range("N2").Value = ProjDB.Range("A" & ProjRow - 1).Value 'Set Previous
Project ID
104 Project_Load
105 End If
106 End With
107 End Sub

```

A

Add, 12, 16, 17
 AddEditClient, 10, 12, 13
 AddEditItem, 11, 14
 Admin, 17, 18, 21
 AdvancedFilter, 12, 14, 16-18, 21, 22
 Application, 17-19, 21, 22
 Apply, 12, 16
 attachments, 17

B

body, 17
 Bold, 21

C

Caller, 19, 21
 CancelBtn_Click, 10, 11
 Category_TabUpdates, 2, 21
 CategoryTab_Select, 21
 CatNumb, 21
 CatQty, 21
 CatRow, 16, 17
 CatWidth, 21
 Cells, 12, 14, 18, 19
 Clear, 12, 16
 ClearContents, 6, 17, 18, 21
 Client_AddEdit, 6, 12
 Client_BuildClientList, 12
 Client_SaveUpdate, 10, 12
 ClientCol, 12
 ClientFld, 12
 ClientRow, 6, 12
 Clients, 6, 12, 17
 Control, 12, 14
 Controls, 12, 14
 CopyToRange, 12, 14, 16-18, 21, 22
 CountA, 21
 CountLarge, 6, 7
 createItem, 17
 CreateObject, 16
 CriteriaRange, 14, 16-18, 21, 22

D

DataOption, 12
 Delay, 18, 19
 Delete, 6, 16
 Dir, 16
 display, 17
 DoEvents, 19

E

Empty, 6, 12, 14, 16, 17, 19, 21
 EntireColumn, 2
 EntireRow, 6, 16
 Explicit, 2-6, 8-12, 14, 16, 21
 ExportAsFixedFormat, 16

F

Field1, 12-14
 FileName, 16, 17
 Fill, 18, 21
 Find, 6, 22
 Font, 21
 Formula, 6, 18, 22

H

Hidden, 2
 Hide, 10-12, 14

I

i, 18, 19
 IDRow, 16
 Intersect, 2, 6, 7
 Item_AddEdit, 6, 14
 Item_SaveUpdate, 11, 14
 ItemCat, 14
 ItemCol, 14
 ItemFld, 14
 ItemRow, 6, 14, 16-19, 21, 22
 Items, 6, 14, 21

K

Key, 12, 16
 Kill, 16

L

LastItemRow, 16, 19
 LastProjItemDBRow, 16-18, 21, 22
 LastResultRow, 12, 16-18, 21, 22
 LastRow, 12, 14, 21
 Left, 7, 17, 21
 LeftPos, 21

M

Max, 22
 Min, 22
 MsgBox, 6, 12, 14, 16, 17, 19, 21, 22
 msoCTrue, 7, 17, 18, 21
 msoFalse, 7, 18, 19, 21
 msoGradientHorizontal, 21

N

NextCat, 17
 NoCatItems, 17
 NoDBItems, 16, 18, 22
 NoItems, 19
 NoSort, 16
 NotSaved, 16

O

Order, 12, 16
 OutApp, 16, 17
 OutMail, 16, 17

P

PageSetup, 17
 Path, 16
 PrintArea, 17
 PrintOut, 18
 Proj_NextProj, 22
 ProjCol, 16, 18, 19
 ProjDB, 16, 18, 19, 22
 ProjDBRow, 16
 Project_CretePDF, 16
 Project_Delete, 16
 Project_Email, 16
 Project_Generate, 16-18
 Project_Load, 6, 17, 22
 Project_New, 16, 18
 Project_PrevProj, 22
 Project_Print, 18
 Project_SavedMsg, 18, 19
 Project_SaveUpdate, 19, 21
 ProjectItemRow, 16
 Projects, 2, 12-14, 16-19, 21, 22
 ProjID, 16, 19
 ProjItemDB, 6, 16-19, 22
 ProjItemDBRow, 6, 16, 19

ProjNumb, 16
 ProjRow, 16, 18, 19, 21, 22

R

Range, 2, 6, 7, 12-14, 16-19, 21, 22
 Replace, 21
 ResultRow, 16, 18, 21, 22
 Row, 6, 7, 12, 14, 16-19, 21, 22

S

SaveBtn_Click, 10, 11
 ScreenUpdating, 17, 18, 21, 22
 SelCat, 16, 19, 21
 SelRow, 14
 SetRange, 12, 16
 Shapes, 7, 17, 18, 21
 Show, 12, 14
 Sort, 12, 16
 SortFields, 12, 16
 SortOn, 12
 StartTime, 18, 19
 Subject, 17

T

Target, 2, 6, 7
 TextFrame2, 21
 TextRange, 21
 ThisWorkbook, 16
 Timer, 19
 Top, 7, 17, 21
 TopPos, 21
 Transparency, 18
 TwoColorGradient, 21
 TypeNumb, 16

U

UCase, 17
 Unique, 12, 14, 16-18, 21, 22
 Unload, 12, 14

V

Value, 2, 6, 7, 12-14, 16-19, 21, 22
 vbDirectory, 16
 vbNo, 16
 vbYes, 6
 vbYesNo, 6, 16
 Visible, 7, 17-19, 21

W

Width, 17, 21
 Worksheet_Change, 2, 6
 Worksheet_SelectionChange, 7
 WorksheetFunction, 21, 22

X

xlAscending, 12
 xlDescending, 16
 xlFilterCopy, 12, 14, 16-18, 21, 22
 xlSortNormal, 12
 xlSortOnValues, 12
 xlTypePDF, 16
 xUp, 12, 14, 16-19, 21, 22
 xlValues, 6, 22
 xlWhole, 6, 22

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