

# VBA SOURCE CODE BOOK

## HOURLY COST CALCULATOR

**EMPLOYEES**

Name	Hourly Cost
Fred Fink	\$30.26
Fred Finklers	\$29.24
James Jimmy	\$33.18
John Smith	\$50.61
Judy Smith	\$36.24
Samantha Kari	\$49.17

**EMPLOYEE DETAILS**

Employee Name: Fred Finklers  
Hourly Amount: \$25.00  
Hours Per Week: 40  
Work Comp. %: 5.40%  
Insurance %: 2.50%

**EMPLOYER PAID TAXES**

Tax Name	Amount
Fed. Inc. Tax	\$2,266.20
FUTA	\$1,499.80
Health Tax	\$84.00
State Unemp. Rate	\$167.00
State SUI	\$2,189.60

**COSTS SUMMARY**

ACTUAL HOURLY COST	ACTUAL HOURLY COST W/ OH	TOTAL ANNUAL COMPENSATION	TOTAL ANNUAL BURDEN
\$39.99	\$42.08	\$64,400	\$79,173

**ANNUAL COMPENSATION VS. EMPLOYER BURDEN**

Total Ann. Burden: \$79,173  
**BURDEN 17%**

**ACTUAL HOURLY COSTS VS. OVERHEAD ALLOCATION**

Hourly Burden: \$2.00  
**% OF OH 5.0%**  
Act. Hourly Cost: \$39.99

**AVERAGE HOURLY COST BY EMPLOYEE TYPE**

Employee Type	Average Hourly Cost
Field Service	39.38
Field Manager	52.26
Office Staff	38.24
Supervisor	50.39
Junior Staff	33.18

**DESIGNED FROM SCRATCH**



*Excel For Freelancers*

## How To Calculate Your Actual Employee Hourly Cost & Labor Burden 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**<sup>®</sup>  
Most Valuable  
Professional



# OUR COURSES & PRODUCTS



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

[Click here to learn more](#)



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

[Click here to learn more](#)

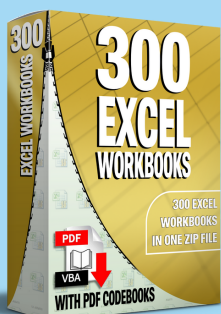
-



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

[Click here to learn more](#)

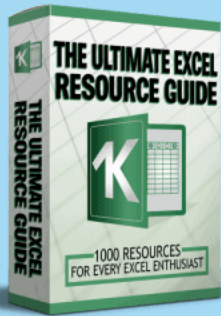
-



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

[Click here to learn more](#)

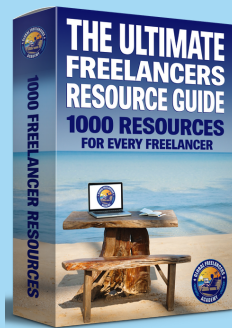
# OUR COURSES & PRODUCTS



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

[Click here to learn more](#)

-



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

[Click here to learn more](#)

-



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

[Click here to learn more](#)

## Table of Contents

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
BenDB.....	3
(Declarations).....	3
EmplDB.....	4
(Declarations).....	4
HourlyCost.....	5
(Declarations).....	5
Worksheet_SelectionChange [Sub ].....	5
PTODB.....	6
(Declarations).....	6
ThisWorkbook.....	7
(Declarations).....	7
Modules.....	8
Empl_Macros.....	8
(Declarations).....	8
Employee_AddNew [Sub ].....	8
Employee_Delete [Sub ].....	8
Employee_ListEmployees [Sub ].....	8
Employee_Load [Sub ].....	9
Employee_SavedMsg [Sub ].....	9
Employee_SaveUpdate [Sub ].....	10

1  
2

Option Explicit

1  
2

Option Explicit

1 Option Explicit

2



```
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("D5:E999" )) Is Nothing And Range("D" & Target.Row).
        Value <> Empty Then
6         Range("B2" ).Value = Range("C" & Target.Row).Value 'Set Empl. ID
7         Range("B5" ).Value = Target.Row 'Select Row
8         Employee_Load 'Run Macro
9     End If
10 End Sub
```

1 Option Explicit

2

1 Option Explicit

2

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, EmplRow As Long, EmplCol As Long, SelRow As
  Long
3
4 Sub Employee_AddNew()
5     With HourlyCost
6         .Range("B2,B5,H4,K4,H6,K6,H8,K4,H10,K10,H12,K12,F16:J999,L16:N999" ).
          ClearContents
7         'Set New Employee Defaults
8         .Range("K4" ).Value = Admin.Range("E5" ).Value 'Employee type
9         .Range("H6" ).Value = Admin.Range("E6" ).Value 'Pay Type
10        .Range("K6" ).Value = Admin.Range("E7" ).Value 'Pay Amount
11        .Range("H8" ).Value = Admin.Range("E8" ).Value 'Workweek Hours
12        .Range("K8" ).Value = Admin.Range("E9" ).Value 'Workweeks per year
13        .Range("H10" ).Value = Admin.Range("E10" ).Value 'Work Comp rate
14        .Range("H4" ).Select
15    End With
16 End Sub
17
18 Sub Employee_Delete()
19     With HourlyCost
20         If MsgBox("Are you sure you want to delete this Employee?" , vbYesNo, "Delete
          Employee" ) = vbNo Then Exit Sub
21         If .Range("B3" ).Value = Empty Then GoTo NotSaved
22         EmplRow = .Range("B3" ).Value 'Employees Row
23         EmplDB.Range(EmplRow & ":" & EmplRow).EntireRow.Delete 'Delete DB Row
24         NotSaved:
25         Employee_AddNew 'Run macro to create new Employees
26         Employee_ListEmployees 'Update Employees list
27     End With
28 End Sub
29
30
31
32
33
34
35
36
37
38 Sub Employee_ListEmployees()
39     HourlyCost.Range("B5,C5:E99" ).ClearContents 'Clear Existing Employees
40     With EmplDB
41         LastRow = .Range("A99999" ).End(xlUp).Row
42         If LastRow < 4 Then Exit Sub
43         .Range("A3:N" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
          "R2:T2" ), Unique:=True
44         LastResultRow = .Range("R99999" ).End(xlUp).Row
45         If LastResultRow < 3 Then Exit Sub
46         If LastResultRow < 4 Then GoTo NoSort 'No sort on only 1 Employee
47         'Sort Alphabetally
48         With .Sort
49             .SortFields.Clear
50             .SortFields.Add Key:=EmplDB.Range("S3" ), SortOn:=xlSortOnValues, Order:=
          xlAscending, DataOption:=xlSortNormal 'Sort
51             .SetRange EmplDB.Range("R3:T" & LastResultRow) 'Set Range
52             .Apply 'Apply Sort
53         End With
54         NoSort:
55         HourlyCost.Range("C5:E" & LastResultRow + 2).Value = .Range("R3:T" & LastRow).
          Value 'Bring Over Employees

```

1 2

```

56 1 2 End With
57
58 With HourlyCost
59 On Error Resume Next
60 SelRow = .Range("C5:C9999").Find(.Range("B2").Value, , xlFormulas, xlWhole).Row
61 If SelRow <> 0 And .Range("B2").Value <> Empty Then .Range("B5").Value = SelRow
62 'Set Selected Employees Row
63 On Error GoTo 0
64 End With
65 End Sub
66
67 Sub Employee_Load()
68 With HourlyCost
69 .Range("H4,K4,H6,K6,H8,K4,H10,K10,H12,K12,F16:J999,L16:N999").ClearContents
70 If .Range("B3").Value = Empty Then
71 MsgBox "Please select a correct Employee to load"
72 Exit Sub
73 End If
74 Application.ScreenUpdating = False
75 EmplRow = .Range("B3").Value 'Employees Row
76 For EmplCol = 2 To 11
77 .Range(EmplDB.Cells(1, EmplCol).Value).Value = EmplDB.Cells(EmplRow, EmplCol).
78 Value 'Bring over data
79 Next EmplCol
80 End With
81 'List Employee Benefits
82 With BenDB
83 LastRow = .Range("A99999").End(xlUp).Row
84 If LastRow < 4 Then GoTo NoBen
85 .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
86 "K2:K3"), CopyToRange:=.Range("O2:R2"), Unique:=True
87 LastResultRow = .Range("O99999").End(xlUp).Row
88 If LastResultRow < 3 Then GoTo NoBen
89 HourlyCost.Range("F16:I" & LastResultRow + 13).Value = .Range("O3:R" &
90 LastResultRow).Value 'Bring over Benefit details
91 End With
92 NoBen:
93 'List Employee Paid Time Off
94 With PTODB
95 LastRow = .Range("A99999").End(xlUp).Row
96 If LastRow < 4 Then GoTo NoPTO
97 .Range("A3:D" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
98 "G2:G3"), CopyToRange:=.Range("J2:L2"), Unique:=True
99 LastResultRow = .Range("J99999").End(xlUp).Row
100 If LastResultRow < 3 Then GoTo NoPTO
101 HourlyCost.Range("L16:N" & LastResultRow + 13).Value = .Range("J3:L" &
102 LastResultRow).Value 'Bring over Benefit details
103 End With
104 NoPTO:
105 Application.ScreenUpdating = True
106 End Sub
107
108 Sub Employee_SavedMsg()
109 With HourlyCost.Shapes("EmplSavedMsg")
110 Dim i As Long, Delay As Double, StartTime As Double
111 .Visible = msoCTrue
112 For i = 1 To 75
113 .Fill.Transparency = i / 75
114 Delay = 0.005
115 StartTime = Timer

```

```

1 2 3
110 Do
111     DoEvents
112     Loop While Timer - StartTime < Delay
113 Next i
114     .Visible = msoFalse
115 End With
116 End Sub

117
118 Sub Employee_SaveUpdate()
119     Dim BenRow As Long, PTORow As Long, BenDBRow As Long, PTODBRow As Long
120     With HourlyCost
121         If .Range("B6").Value < 6 Then 'Empty Fields
122             MsgBox "Please make sure to fill in all required fields before saving"
123             Exit Sub
124         End If
125         If .Range("B3").Value = Empty Then 'New Employee
126             EmplRow = EmplDB.Range("A9999").End(xlUp).Row + 1 'First Avail Row
127             .Range("B2").Value = .Range("B4").Value 'New Employee ID
128             EmplDB.Range("A" & EmplRow).Value = .Range("B2").Value 'Set Employees ID
129         Else 'Existing Employees
130             EmplRow = .Range("B3").Value 'Existing Employees Row
131         End If
132         For EmplCol = 2 To 15
133             EmplDB.Cells(EmplRow, EmplCol).Value = .Range(EmplDB.Cells(1, EmplCol).Value).
134                 Value 'Save/Update Data
135         Next EmplCol
136         'Save/Update Benefits
137         LastRow = .Range("G999").End(xlUp).Row
138         If LastRow < 16 Then GoTo NoBen
139         For BenRow = 16 To LastRow
140             If .Range("F" & BenRow).Value = Empty Then 'Not Prev. saved
141                 BenDBRow = BenDB.Range("A9999").End(xlUp).Row + 1
142                 BenDB.Range("A" & BenDBRow).Value = .Range("B2").Value 'Employee ID
143                 BenDB.Range("F" & BenDBRow).Value = "=Row()" 'Set DB Row
144                 .Range("F" & BenRow).Value = BenDBRow
145             Else 'Existing Benefit
146                 BenDBRow = .Range("F" & BenRow).Value 'Ben DB Row
147             End If
148             BenDB.Range("B" & BenDBRow & ":D" & BenDBRow).Value = .Range("G" & BenRow &
149                 ":I" & BenRow).Value 'Name, Amount, Freq.
150             BenDB.Range("E" & BenDBRow).Value = .Range("K" & BenRow).Value 'Annual Cost
151         Next BenRow
152         NoBen:
153         'Save/Update PTO
154         LastRow = .Range("M999").End(xlUp).Row
155         If LastRow < 16 Then GoTo NoPTO
156         For PTORow = 16 To LastRow
157             If .Range("L" & PTORow).Value = Empty Then 'Not Saved
158                 PTODBRow = PTODB.Range("A9999").End(xlUp).Row + 1
159                 PTODB.Range("A" & PTODBRow).Value = .Range("B2").Value 'Employee ID
160                 PTODB.Range("D" & PTODBRow).Value = "=Row()" 'Set DB Row
161                 .Range("L" & PTORow).Value = PTODBRow
162             Else 'Existing PTO
163                 PTODBRow = .Range("L" & PTORow).Value
164             End If
165             PTODB.Range("B" & PTODBRow & ":C" & PTODBRow).Value = .Range("M" & PTORow &
166                 ":N" & PTORow).Value 'Name, Amount, Freq.
167         Next PTORow
168         NoPTO:
169         Employee_ListEmployees 'Update Employees list

```

```
167 | 1 2 Employee_SavedMsg  
168 | | End With  
169 | | End Sub
```

**A**

Add, 8  
Admin, 8  
AdvancedFilter, 8, 9  
Application, 9  
Apply, 8

**B**

BenDB, 9, 10  
BenDBRow, 10  
BenRow, 10

**C**

Cells, 9, 10  
Clear, 8  
ClearContents, 8, 9  
CopyToRange, 8, 9  
CountLarge, 5  
CriteriaRange, 9

**D**

DataOption, 8  
Delay, 9, 10  
Delete, 8  
DoEvents, 10

**E**

EmplCol, 8-10  
EmplDB, 8-10  
Employee\_AddNew, 8  
Employee\_Delete, 8  
Employee\_ListEmployees, 8, 10  
Employee\_Load, 5, 9  
Employee\_SavedMsg, 9, 11  
Employee\_SaveUpdate, 10  
EmplRow, 8-10  
Empty, 5, 8-10  
EntireRow, 8  
Explicit, 2-8

**F**

Fill, 9  
Find, 9

**H**

HourlyCost, 8-10

**I**

i, 9, 10  
Intersect, 5

**K**

Key, 8

**L**

LastResultRow, 8, 9  
LastRow, 8-10

**M**

MsgBox, 8-10  
msoCTrue, 9  
msoFalse, 10

**N**

NoBen, 9, 10  
NoPTO, 9, 10  
NoSort, 8  
NotSaved, 8

**O**

Order, 8

**P**

PTODB, 9, 10  
PTODBRow, 10  
PTORow, 10

**R**

Range, 5, 8-10  
Row, 5, 8-10

**S**

ScreenUpdating, 9  
SelRow, 8, 9  
SetRange, 8  
Shapes, 9  
Sort, 8  
SortFields, 8  
SortOn, 8  
StartTime, 9, 10

**T**

Target, 5  
Timer, 9, 10  
Transparency, 9

**U**

Unique, 8, 9

**V**

Value, 5, 8-10  
vbNo, 8  
vbYesNo, 8  
Visible, 9, 10

**W**

Worksheet\_SelectionChange, 5

**X**

xlAscending, 8  
xlFilterCopy, 8, 9  
xlFormulas, 9  
xlSortNormal, 8  
xlSortOnValues, 8  
xlUp, 8-10  
xlWhole, 9



# 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*