

# VBA SOURCE CODE BOOK



Learn How To Use These 7  
Tricks To Speed Up Your  
Slow VBA Code In Excel



[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 nearly 300,000 YouTube subscribers, 14,000,000 video views, 200+ comprehensive training videos, and a thriving 40,000 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](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



**Microsoft®**  
Most Valuable  
Professional



# OUR COURSES & PRODUCTS



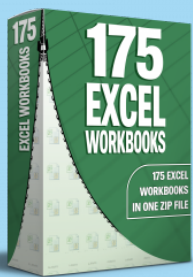
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[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 175 of my BEST Applications into a SINGLE ZIP File which also includes the "175 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](#)

## Table of Contents

Projects.....	2
VBAProject .....	2
Documents .....	2
Sheet1.....	2
(Declarations).....	2
Sheet3.....	4
(Declarations).....	4
ThisWorkbook .....	6
(Declarations).....	6
Modules.....	8
CodeRunTime .....	8
(Declarations).....	8
CalculateRunTime_Minutes [Sub ] .....	8
StartClock [Sub ] .....	8
StopClock [Sub ] .....	8
ProductMacros .....	10
(Declarations).....	10
UpdateProducts [Sub ] .....	10

1 Option Explicit

2

**E**

Explicit, 2

1 Option Explicit

2

**E**

Explicit, 4



1 Option Explicit  
2

**E**

Explicit, 6

```
1 Option Explicit
2 Dim StartTime As Double
3 Dim SecondsElapsed As Double
4
5 Sub CalculateRunTime_Minutes()
6     Dim StartTime As Double
7     Dim MinutesElapsed As String
8
9     'Remember time when macro starts
10    StartTime = Timer
11    'Determine how many seconds code took to run
12    MinutesElapsed = Format((Timer - StartTime) / 86400, "hh:mm:ss" )
13
14    'Notify user in seconds
15    MsgBox "This code ran successfully in " & MinutesElapsed & " minutes" , vbInformation
16 End Sub
17 Sub StartClock()
18     'Remember time when macro starts
19     StartTime = Timer
20 End Sub
21
22 Sub StopClock()
23     SecondsElapsed = Round(Timer - StartTime, 2) 'Determine how many seconds code took to run
24     MsgBox "This code ran successfully in " & SecondsElapsed & " seconds" , vbInformation
25     'Notify user in seconds
26 End Sub
```

**C**

CalculateRunTime\_Minutes, 8

**E**

Explicit, 8

**F**

Format, 8

**M**

MinutesElapsed, 8

MsgBox, 8

**R**

Round, 8

**S**

SecondsElapsed, 8

StartClock, 8

StartTime, 8

StopClock, 8

**T**

Timer, 8

**V**

vbInformation, 8

```

1  Option Explicit
2
3  Sub UpdateProducts()
4
5      StartClock
6      With Application
7          .Calculation = xlCalculationManual
8          .ScreenUpdating = False
9          .EnableEvents = False
10     End With
11
12     Dim ChangeNumber As Double
13     Dim LastProdRow As Long
14     Dim ProdRow As Long
15     Dim ProdType As String
16     Dim LastResultRow As Long
17     Dim ResultRow As Long
18     Dim ChangeDirection As String
19
20     With Sheet3
21         LastProdRow = .Range("A999999").End(xlUp).Row 'Last Product Row
22         ChangeNumber = Sheet1.Range("E4").Value 'Change Number
23         ProdType = Sheet1.Range("C4").Value 'Product Type
24         ChangeDirection = Sheet1.Range("G4").Value 'Change Direction
25         If ChangeDirection = "Decrease" Then ChangeNumber = -ChangeNumber
26
27         .Range("P3:V999999").ClearContents
28         If ProdType = "[All Types]" Then 'On All Types
29             On Error Resume Next
30             .Names("Criteria").Delete
31             On Error GoTo 0
32             .Range("A2:G" & LastProdRow).AdvancedFilter xlFilterCopy, CopyToRange:=.Range(
33                 "P2:V2"), Unique:=False
34         Else: 'On Specific Types
35             .Range("A2:G" & LastProdRow).AdvancedFilter xlFilterCopy, .Range("N2:N3"),
36                 CopyToRange:=.Range("P2:V2"), Unique:=False
37         End If
38
39         LastResultRow = .Range("P999999").End(xlUp).Row 'Last Results Row
40         If LastResultRow < 3 Then GoTo NoData
41
42         For ResultRow = 3 To LastResultRow
43             ProdRow = .Range("V" & ResultRow).Value 'Product Row
44             If Sheet1.Range("F4").Value = "$" Then
45                 .Range("D" & ProdRow).Value = .Range("D" & ProdRow).Value + ChangeNumber '
46                 Update Sales Price
47             Else: 'If Change by %
48                 .Range("D" & ProdRow).Value = .Range("D" & ProdRow).Value + ((ChangeNumber /
49                     100) * .Range("D" & ProdRow).Value)
50             End If
51         Next ResultRow
52     End With
53     NoData:
54     With Application
55         .Calculation = xlCalculationAutomatic
56         .ScreenUpdating = True
57         .EnableEvents = True
58     End With
59     StopClock
60 End Sub

```

58

59

60

## A

AdvancedFilter, [10](#)  
Application, [10](#)

## C

Calculation, [10](#)  
ChangeDirection, [10](#)  
ChangeNumber, [10](#)  
ClearContents, [10](#)  
CopyToRange, [10](#)

## D

Delete, [10](#)

## E

EnableEvents, [10](#)  
Explicit, [10](#)

## L

LastProdRow, [10](#)  
LastResultRow, [10](#)

## N

Names, [10](#)  
NoData, [10](#)

## P

ProdRow, [10](#)  
ProdType, [10](#)

## R

Range, [10](#)  
ResultRow, [10](#)  
Row, [10](#)

## S

ScreenUpdating, [10](#)  
Sheet1, [10](#)  
Sheet3, [10](#)  
StartClock, [10](#)  
StopClock, [10](#)

## U

Unique, [10](#)  
UpdateProducts, [10](#)

## V

Value, [10](#)

## X

xlCalculationAutomatic, [10](#)  
xlCalculationManual, [10](#)  
xlFilterCopy, [10](#)  
xlUp, [10](#)

# 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*