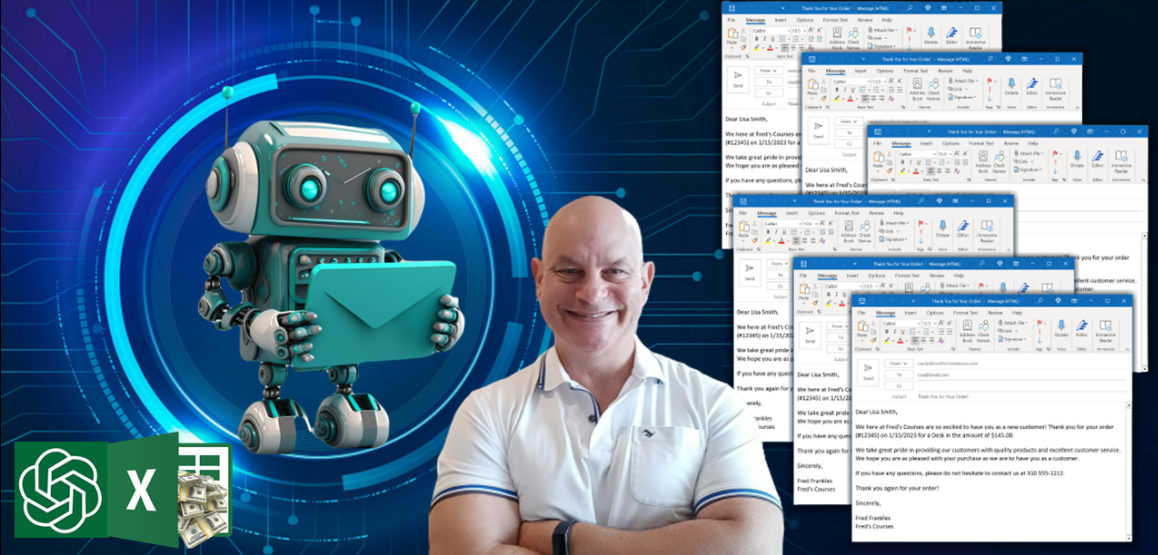


VBA SOURCE CODE BOOK

ChatGPT EMAIL GENERATOR



 *Excel For Freelancers*

**How To Create Professional
Emails In 1 Click With
ChatGPT & AI 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 over 442,000 YouTube subscribers, 33,000,000 video views, 250+ comprehensive training videos, and a thriving 60,000 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



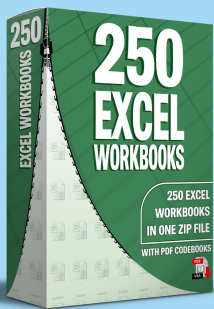
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



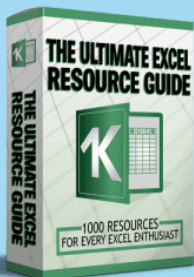
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[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](#)



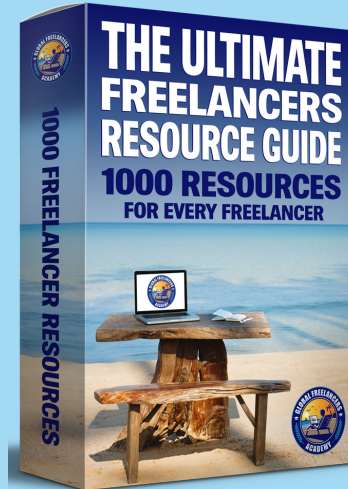
With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)



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 along with a ton of resources for anyone to reach their goals through freelancing.

[Click here to learn more](#)



I've put together all the 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](#)

Table of Contents

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Worksheet_Change [Sub].....	2
EmailGen.....	3
(Declarations).....	3
Worksheet_Change [Sub].....	3
Worksheet_SelectionChange [Sub].....	3
EmailLog.....	4
(Declarations).....	4
Prompts.....	5
(Declarations).....	5
ThisWorkbook.....	6
(Declarations).....	6
Modules.....	7
EmailPrompt_Macros.....	7
(Declarations).....	7
Prompt_AddNew [Sub].....	7
Prompt_Delete [Sub].....	7
Prompt_Duplicate [Sub].....	7
Prompt_ListEmailNames [Sub].....	7
Prompt_ListEmailTypes [Sub].....	7
Prompt_Load [Sub].....	8
Prompt_SaveUpdate [Sub].....	8
EmailSend_Macros.....	9
(Declarations).....	9
Create_Email [Sub].....	9
Generate_Response [Sub].....	9
UpdatePrömt [Sub].....	10

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     'Update Prompt on Variable Change
5     If Not Intersect(Target, Range("D5:D7" )) Is Nothing And Target.Value <> Empty Then
6         UpdatePromt
7     End Sub
```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     'Update Prompt on Variable Change
5     If Not Intersect(Target, Range("H10,I7,K7,M7,O7,O5,R5,R7,T7,V7,X7" )) Is Nothing And
6         Range("B2" ).Value = False Then UpdatePromt
7 End Sub
8
9 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
10     If Target.CountLarge > 1 Then Exit Sub
11     'On Selection Of Email Type
12     If Not Intersect(Target, Range("D5:D999" )) Is Nothing And Range("D" & Target.Row).
13         Value <> Empty Then
14         Range("B7" ).Value = Target.Row
15         Prompt_ListEmailNames
16     End If
17     'On Selection Of Email Prompts
18     If Not Intersect(Target, Range("E5:E999" )) Is Nothing And Range("E" & Target.Row).
19         Value <> Empty Then
20         Range("B3" ).Value = Range("F" & Target.Row).Value 'Load Prompt ID
21         Range("B8" ).Value = Target.Row
22         Prompt_Load 'Run Macro To load Prompt
23     End If
24 End Sub
```

1 Option Explicit

2

1
2

Option Explicit

1 Option Explicit

2

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, PromptRow As Long, PromptCol As Long
3
4
5 Sub Prompt_AddNew()
6     EmailGen.Range("B2" ).Value = True
7     EmailGen.Range(
8         "B3,B7:B8,I5:J5,L5:M5,O5,R5:T5,V5:X5,I7,K7,M7,O7,R7,T7,V7,X7,H10:O19,H23:O32,R10:X10,
9         Q12:X38" ).ClearContents 'Clear Fields
10    EmailGen.Range("B2" ).Value = False
11 End Sub
12
13 Sub Prompt_Delete()
14     Dim EmailRow As Long
15     If MsgBox("Are you sure you want to delete this prompt?" , vbYesNo, "Delete Prompt"
16         ) = vbNo Then Exit Sub
17     If EmailGen.Range("B4" ).Value = "" Then GoTo NotSaved
18     EmailRow = EmailGen.Range("B4" ).Value 'Email Row
19     Prompts.Range(EmailRow & ":" & EmailRow).EntireRow.Delete
20     NotSaved:
21     Prompt_AddNew
22     Prompt_ListEmailTypes 'Run Macro To List Email Types
23 End Sub
24
25 Sub Prompt_Duplicate()
26     EmailGen.Range("B3" ).ClearContents
27     EmailGen.Range("L5" ).Value = EmailGen.Range("L5" ).Value & "-Copy"
28     Prompt_SaveUpdate
29 End Sub
30
31 Sub Prompt_ListEmailNames()
32     If EmailGen.Range("B7" ).Value = Empty Then Exit Sub 'Exit on No Selected Email
33     Types
34     EmailGen.Range("B8,E5:F999" ).ClearContents 'Clear Existing Email Names & Prompt
35     ID's
36     With Prompts
37         LastRow = .Range("A99999" ).End(xlUp).Row 'Last Row
38         If LastRow < 4 Then Exit Sub
39         .Range("A3:C" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
40             "M2:M3" ), CopyToRange:=.Range("P2:Q2" ), Unique:=True
41         LastResultRow = .Range("P99999" ).End(xlUp).Row
42         If LastResultRow < 3 Then Exit Sub
43         EmailGen.Range("E5:F" & LastResultRow + 2).Value = .Range("P3:Q" &
44             LastResultRow).Value 'Bring Over Email Names & Prompt ID's
45         On Error Resume Next
46         EmailGen.Range("B8" ).Value = EmailGen.Range("E5:E" & LastResultRow + 2).Find(
47             EmailGen.Range("L5" ).Value, , xlValues, xlWhole).Row 'Set Selected Email Name
48             Row (If Any)
49         On Error GoTo 0
50     End With
51 End Sub
52
53 Sub Prompt_ListEmailTypes()
54     'Get Unique Email Types
55     EmailGen.Range("B7:B8,D5:F999" ).ClearContents 'Clear Existing Email Types & Names
56     With Prompts
57         LastRow = .Range("A99999" ).End(xlUp).Row 'Last Row
58         If LastRow < 4 Then Exit Sub
59         On Error Resume Next
60         .Names("Criteria" ).Delete 'Delete any existing criteria (important for advanced

```

1 2

```

1 2
filters that do not have criteria)
53 On Error GoTo 0
54 .Range("B3:B" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
"O2" ), Unique:=True
55 LastResultRow = .Range("O99999" ).End(xlUp).Row
56 If LastResultRow < 3 Then Exit Sub
57 EmailGen.Range("D5:D" & LastResultRow + 2).Value = .Range("O3:O" &
LastResultRow).Value 'Bring Over Email Types
58 On Error Resume Next
59 EmailGen.Range("B7" ).Value = EmailGen.Range("D5:D" & LastResultRow + 2).Find(
EmailGen.Range("I5" ).Value, , xlValues, xlWhole).Row 'Set Selected Email Type
Row (If Any)
60 On Error GoTo 0
61 End With
62 End Sub

63 Sub Prompt_Load()
64 With EmailGen
65 .Range(
"I5:J5,L5:M5,O5,R5:T5,V5:X5,I7,K7,M7,O7,R7,T7,V7,X7,H10:O19,H23:O32,R10:X10,Q12:X3
8" ).ClearContents 'Clear Fields
66 If .Range("B4" ).Value = Empty Then
67 MsgBox "Please select an email prompt to view"
68 Exit Sub
69 End If
70 PromptRow = .Range("B4" ).Value
71 .Range("B2" ).Value = True 'Set Email Load to True
72 For PromptCol = 2 To 11
73 .Range(Prompts.Cells(1, PromptCol).Value).Value = Prompts.Cells(PromptRow,
PromptCol).Value 'Load Data in Sheet Form
74 Next PromptCol
75 .Range("B2" ).Value = False 'Set Email Load to False
76 End With
77 UpdatePrompt 'Run Macro To Load Prompt
78 End Sub

79

80 Sub Prompt_SaveUpdate()
81 With EmailGen
82 If .Range("I5" ).Value = Empty Or .Range("L5" ).Value = Empty Then
83 MsgBox "Please make sure to include an Email Type and Email Name. These
cannot be left blank"
84 Exit Sub
85 End If
86
87 If .Range("B4" ).Value = "" Then 'New Email Prompt
88 PromptRow = Prompts.Range("A9999" ).End(xlUp).Row + 1 'First Avail Row
89 .Range("B3" ).Value = .Range("B5" ).Value 'Next Prompt ID
90 Prompts.Range("A" & PromptRow).Value = .Range("B3" ).Value 'Set Prompt ID
91 Else 'Existing Email Prompt
92 PromptRow = .Range("B4" ).Value
93 End If
94 For PromptCol = 2 To 11
95 Prompts.Cells(PromptRow, PromptCol).Value = .Range(Prompts.Cells(1, PromptCol)
.Value).Value 'Save/Update data
96 Next PromptCol
97 Prompts.Range(PromptRow & ":" & PromptRow).WrapText = False
98 MsgBox "Email Prompt Saved"
99 End With
100 Prompt_ListEmailTypes 'Run Macro To List Email Types
101 Prompt_ListEmailNames 'Run Macro to list email names
102 End Sub

```

```

1 Option Explicit
2 Sub Create_Email()
3     Dim OutApp As Object, OutMail As Object
4     Dim DispEmail As String, EmailType As String, Subject As String, Message As String,
5     EmailTo As String
6     Dim EmailRow As Long
7     With EmailGen
8         EmailTo = .Range("V5" ).Value 'Email to
9         Subject = .Range("R10" ).Value 'Subject
10        Message = .Range("Q12" ).Value 'Message
11    End With
12    DispEmail = Admin.Range("G5" ).Value 'Email Option (Yes/no)
13    Set OutApp = CreateObject("Outlook.Application" )
14    Set OutMail = OutApp.CreateItem(0)
15    With OutMail
16        .To = EmailTo
17        .Subject = Subject
18        .Body = Message
19        If DispEmail = "Yes" Then .Display Else .Send
20    End With
21    With EmailLog
22        EmailRow = .Range("A99999" ).End(xlUp).Row + 1
23        .Range("A" & EmailRow).Value = Now 'Set Day and time
24        .Range("B" & EmailRow).Value = EmailTo 'Sent To
25        .Range("C" & EmailRow).Value = Subject 'Email Subject
26        .Range("D" & EmailRow).Value = Message ' Email Message
27        .Range(EmailRow & ":" & EmailRow).WrapText = False
28    End With
29 End Sub
30
31 Sub Generate_Response()
32     Dim oXMLHTTP As Object
33     Dim Response As String, AIBody As String, Prompt As String, SecretKey As String,
34     ParsedAnswer As String, Subject As String
35     Dim URL As String, SendTo As String, Sitelink As String, Model As String, Temp As
36     String, EmailType As String
37     UpdatePromt 'Run Macro to Update prompt
38     'Determine API Request Variables
39     With Admin
40         EmailType = .Range("G4" ).Value 'Email Type
41         SecretKey = .Range("G6" ).Value 'Secret Key
42         Model = .Range("G7" ).Value 'Model
43         Temp = .Range("G8" ).Value 'Temperature
44     End With
45     If SecretKey = "" Then
46         MsgBox "Please make sure to enter a secret key on the Admin sheet"
47         Exit Sub
48     End If
49     With EmailGen
50         Prompt = Replace(Replace(Replace(.Range("H23" ).Value, Chr(34), "\" & Chr(34)),
51         vbNewLine, "\n" ), Chr(10), "\n" ) 'Prompt
52         If Prompt = "" Then
53             MsgBox "Please make sure to enter a prompt"
54             .Range("H10" ).Select
55             Exit Sub
56         End If
57         .Shapes("GenTextMsg" ).Visible = msoCTrue 'Display Generating Response Shape
58         Application.ScreenUpdating = True
59         DoEvents
60         Set oXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP" )

```

1 2

```

1 2
58 oXMLHTTP.Open "POST" , "https://api.openai.com/v1/completions" , False 'Set Open
Posts For Text
59 AIBody = "{" "model" ": " "" & Model & "" ", " "prompt" ": " "" & Prompt & ""
", " "temperature" ": " & Temp & " , " "max_tokens" ": 3500}" 'Request Body
For Text Response
60 Debug.Print AIBody
61 oXMLHTTP.SetRequestHeader "Content-Type" , "application/json" 'Set Header Type
62 oXMLHTTP.SetRequestHeader "Authorization" , "Bearer " & SecretKey 'Set Header
Authorization
63 oXMLHTTP.Send AIBody
64
65 Response = oXMLHTTP.responseText 'Extract Response
66 Debug.Print Response
67 If InStr(Response, "Incorrect API key provided" ) > 0 Then
68     MsgBox "Incorrect API key provided. You can find your API key at " & vbCrLf &
" https://platform.openai.com/account/api-keys."
69     Exit Sub
70 End If
71 If InStr(Response, "maximum context length" ) > 0 Then
72     MsgBox "The length of the prompt and response is too long, please break it up
into smaller parts or reduce the temperature"
73     Exit Sub
74 End If
75 ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "[" & Chr(34)
& "text" & Chr(34) & ":" & Chr(34)) - 9) 'Remove First part of coded response
76 ParsedAnswer = Right(ParsedAnswer, Len(ParsedAnswer) - 4) 'Remove First two
empty lines
77 ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(34)
& "index" ) - 1) 'Remove last part of coded response
78 ParsedAnswer = Replace(Replace(ParsedAnswer, "\n" , Chr(10)), "\" & Chr(34), Chr
(34)) 'Replace new lines with carriage return characters and quotation marks
79
80 'Extract Subject
81 If InStr(ParsedAnswer, "Subject Line:" ) > 0 Then 'If Response contains 'Subject
Line
82     Subject = Replace(Left(ParsedAnswer, InStr(ParsedAnswer, Chr(10)) - 1),
"Subject Line: " , "" )
83     ParsedAnswer = Replace(ParsedAnswer, "Subject Line: " & Subject & Chr(10),
"" )
84 Else 'If Response only contains 'Subject:'
85     Subject = Replace(Left(ParsedAnswer, InStr(ParsedAnswer, Chr(10)) - 1),
"Subject: " , "" )
86     ParsedAnswer = Replace(ParsedAnswer, "Subject: " & Subject & Chr(10), "" )
87 End If
88
89
90 If Left(ParsedAnswer, 1) = Chr(10) Then ParsedAnswer = Right(ParsedAnswer, Len(
ParsedAnswer) - 1) 'Remove any beginning carriage returns if they exist
91 .Range("R10" ).Value = Subject
92 .Range("Q12" ).Value = ParsedAnswer
93 .Shapes("GenTextMsg" ).Visible = msoFalse 'Hide Information text shape
94 End With
95 End Sub
96 Sub UpdatePromt()
97 Dim Prompt As String, SentFrom As String, CompName As String, CompContact As String
98 Dim VarName1 As String, VarName2 As String, VarName3 As String, VarName4 As String
99 Dim VarValue1 As String, VarValue2 As String, VarValue3 As String, VarValue4 As
String
100 Dim SendTo As String, Tone As String
101 SentFrom = Admin.Range("D5" ).Value 'Sent From Name

```

```
1
102 CompName = Admin.Range("D6").Value 'Sent From Company
103 CompContact = Admin.Range("D7").Value 'Company Contact Info
104 With EmailGen
105     Tone = .Range("O5").Value 'Email Tone
106     SendTo = .Range("R5").Value 'Send to Name
107     Prompt = .Range("H10").Value 'Prompt With Variables
108     VarName1 = "[" & .Range("I7").Value & "]" 'Variable Name 1
109     VarName2 = "[" & .Range("K7").Value & "]" 'Variable Name 2
110     VarName3 = "[" & .Range("M7").Value & "]" 'Variable Name 3
111     VarName4 = "[" & .Range("O7").Value & "]" 'Variable Name 4
112     VarValue1 = .Range("R7").Value 'Variable Value 1
113     VarValue2 = .Range("T7").Value 'Variable Value 2
114     VarValue3 = .Range("V7").Value 'Variable Value 3
115     VarValue4 = .Range("X7").Value 'Variable Value 4
116
117     'Update Formats if not General
118     If .Range("R7").NumberFormat <> "General" Then VarValue1 = Format(VarValue1, .
Range("R7").NumberFormat)
119     If .Range("T7").NumberFormat <> "General" Then VarValue2 = Format(VarValue2, .
Range("T7").NumberFormat)
120     If .Range("V7").NumberFormat <> "General" Then VarValue3 = Format(VarValue3, .
Range("V7").NumberFormat)
121     If .Range("X7").NumberFormat <> "General" Then VarValue4 = Format(VarValue4, .
Range("X7").NumberFormat)
122
123     'Replace from Email Settings
124     Prompt = Replace(Replace(Replace(Replace(Replace(Prompt, "[Tone]", Tone),
"[Send To]", SendTo), "[Sent From]", SentFrom), "[Company Name]", CompName),
"[Company Contact]", CompContact)
125     'Replace variables with values
126     .Range("H23").Value = Replace(Replace(Replace(Replace(Prompt, VarName1,
VarValue1), VarName2, VarValue2), VarName3, VarValue3), VarName4, VarValue4)
127 End With
128 End Sub
```

A

Admin, 9-11
 AdvancedFilter, 7, 8
 AllBody, 9, 10
 Application, 9

B

Body, 9

C

Cells, 8
 Chr, 9, 10
 ClearContents, 7, 8
 CompContact, 10, 11
 CompName, 10, 11
 CopyToRange, 7, 8
 CountLarge, 3
 Create_Email, 9
 CreateItem, 9
 CreateObject, 9
 CriteriaRange, 7

D

Debug, 10
 Delete, 7
 DispEmail, 9
 Display, 9
 DoEvents, 9

E

EmailGen, 7-9, 11
 EmailLog, 9
 EmailRow, 7, 9
 EmailTo, 9
 EmailType, 9
 Empty, 2, 3, 7, 8
 EntireRow, 7
 Explicit, 2-7, 9

F

Find, 7, 8
 Format, 11

G

Generate_Response, 9

I

InStr, 10
 InStrRev, 10
 Intersect, 2, 3

L

LastResultRow, 7, 8
 LastRow, 7, 8
 Left, 10
 Len, 10

M

Message, 9
 Model, 9, 10
 MsgBox, 7-10
 msoCTrue, 9
 msoFalse, 10

N

Names, 7
 NotSaved, 7
 Now, 9
 NumberFormat, 11

O

Open, 10
 OutApp, 9
 OutMail, 9
 oXMLHTTP, 9, 10

P

ParsedAnswer, 9, 10
 Print, 10
 Prompt, 9-11
 Prompt_AddNew, 7
 Prompt_Delete, 7
 Prompt_Duplicate, 7
 Prompt_ListEmailNames, 3, 7, 8
 Prompt_ListEmailTypes, 7, 8
 Prompt_Load, 3, 8
 Prompt_SaveUpdate, 7, 8
 PromptCol, 7, 8
 PromptRow, 7, 8
 Prompts, 7, 8

R

Range, 2, 3, 7-11
 Replace, 9-11
 Response, 9, 10
 responseText, 10
 Right, 10
 Row, 3, 7-9

S

ScreenUpdating, 9
 SecretKey, 9, 10
 Send, 9, 10
 SendTo, 9-11
 SentFrom, 10, 11
 SetRequestHeader, 10
 Shapes, 9, 10
 SiteLink, 9
 Subject, 9, 10

T

Target, 2, 3
 Temp, 9, 10
 Tone, 10, 11

U

Unique, 7, 8
 UpdatePromt, 2, 3, 8-10
 URL, 9

V

Value, 2, 3, 7-11
 VarName1, 10, 11
 VarName2, 10, 11
 VarName3, 10, 11
 VarName4, 10, 11
 VarValue1, 10, 11
 VarValue2, 10, 11
 VarValue3, 10, 11
 VarValue4, 10, 11
 vbCrLf, 10
 vbNewLine, 9
 vbNo, 7
 vbYesNo, 7
 Visible, 9, 10

W

Worksheet_Change, 2, 3
 Worksheet_SelectionChange, 3

WrapText, 8, 9

X

xlFilterCopy, 7, 8
 xlUp, 7-9
 xlValues, 7, 8
 xlWhole, 7, 8

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