

VBA SOURCE CODE BOOK

DYNAMIC POP-UP HELP

**FREE
TEMPLATE
DOWNLOAD**

USER HELP ITEM

Worksheet: CustMar
Cell Address: \$E\$12
Help Text

SAVE CANCEL



Excel For Freelancers

How To Add Help Pop-Ups To Any Excel Workbook With This New System



**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[®]
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](#)

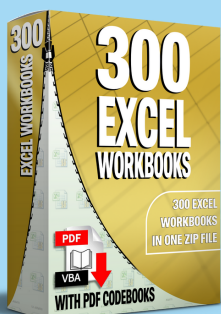
-



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](#)

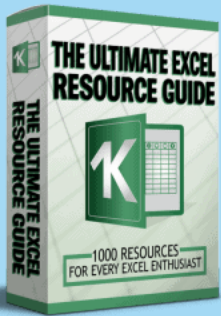
-



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](#)

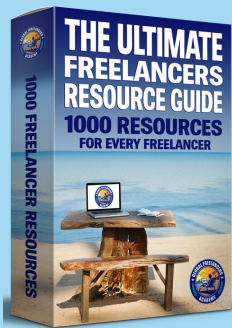
OUR COURSES & PRODUCTS



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](#)

Projects.....	3
VBAProject.....	3
Documents.....	3
Contacts.....	3
(Declarations).....	3
CustMgr.....	4
(Declarations).....	4
Worksheet_Change [Sub].....	4
Worksheet_SelectionChange [Sub].....	4
Customers.....	5
(Declarations).....	5
HelpData.....	6
(Declarations).....	6
Invoice.....	7
(Declarations).....	7
Worksheet_Change [Sub].....	7
Worksheet_SelectionChange [Sub].....	7
InvoiceItems.....	8
(Declarations).....	8
InvoiceList.....	9
(Declarations).....	9
Items.....	10
(Declarations).....	10
ThisWorkbook.....	11
(Declarations).....	11
Workbook_Open [Sub].....	11
Workbook_SheetBeforeRightClick [Sub].....	11
Workbook_WindowDeactivate [Sub].....	11
WorkOrders.....	12
(Declarations).....	12
Forms.....	13
HelpFrm.....	13
(Declarations).....	13
CancelBtn_Click [Sub].....	13
SaveBtn_Click [Sub].....	13
Modules.....	14
Contact_Macros.....	14
(Declarations).....	14
Contact_AddNew [Sub].....	14
Contact_Delete [Sub].....	14
Contact_Load [Sub].....	14
Contact_SaveNew [Sub].....	14
Cust_LoadMac.....	15
(Declarations).....	15
Customer_Load [Sub].....	15
LoadContacts [Sub].....	15
Cust_MiscMacros.....	16
(Declarations).....	16
Customer_AddNew [Sub].....	16
Customer_CancelNew [Sub].....	16
Customer_Delete [Sub].....	16
Cust_SaveMac.....	17
(Declarations).....	17
Customer_SaveUpdate [Sub].....	17
Help_Macros.....	18
(Declarations).....	18
AddEditHelp [Sub].....	18
CheckForHelp [Sub].....	18
CloseHelp [Sub].....	19
DeleteHelp [Sub].....	19
Rightclick_Cancel [Sub].....	19
RightClick_ShowButtons [Sub].....	19
SaveHelp [Sub].....	20
ShowHelpPopUp [Sub].....	20
InvoiceMacros.....	22
(Declarations).....	22

Table of Contents

Invoice_CancelNew [Sub]	22
Invoice_Load [Sub]	22
Invoice_New [Sub]	22
Invoice_Print [Sub]	22
Invoice_SaveUpdate [Sub]	22

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F2" )) Is Nothing And Range("B10" ).Value <> Empty
5         Then Customer_Load
6     If Not Intersect(Target, Range("N4:N12" )) Is Nothing And Range("B13" ).Value =
7         False And Range("B12" ).Value <> Empty Then
8         Dim ContDbRow As Long
9         ContDbRow = Range("R" & Range("B12" ).Value).Value
10        Contacts.Cells(ContDbRow, Target.Offset(0, 1).Value).Value = Target.Value
11        LoadContacts 'Reload Contacts
12    End If
13 End Sub
14
15 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
16     CheckForHelp
17     If Not Intersect(Target, Range("M16:N29" )) Is Nothing And Range("M" & Target.Row).
18         Value <> Empty Then
19         Range("B12" ).Value = Target.Row
20         Contact_Load
21     End If
22 End Sub
```


1 Option Explicit
2

1
2

Option Explicit

```
1 Option Explicit
```

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

```
4 'On Customer Name Change
```

```
5 If Not Intersect(Target, Range("E3" )) Is Nothing Then
```

```
6 If Target.Value <> Empty And Range("B2" ).Value <> Empty Then
```

```
7 Dim CustRow As Long
```

```
8 CustRow = Range("B2" ).Value 'Customer Row
```

```
9 Range("E4" ).Value = Customers.Range("I" & CustRow).Value 'Address 1
```

```
10 Range("E5" ).Value = Customers.Range("J" & CustRow).Value & ", " _
```

```
11 & Customers.Range("K" & CustRow).Value & " " _
```

```
12 & Customers.Range("L" & CustRow).Value 'Address 2
```

```
13 End If
```

```
14 End If
```

```
15  
16 'On Invoice Item Name Change (But not on Invoice Load)
```

```
17 If Not Intersect(Target, Range("D8:D27" )) Is Nothing And Range("B5" ).Value = False  
18 Then
```

```
19 If Range("C" & Target.Row).Value <> Empty Then
```

```
20 Dim ItemRow As Long
```

```
21 ItemRow = Range("C" & Target.Row).Value 'Item Row
```

```
22 Range("E" & Target.Row & ":G" & Target.Row).Value = Items.Range("B" &
```

```
23 ItemRow & ":D" & ItemRow).Value 'Copy over Desc, Qty, Price
```

```
24 End If
```

```
25 End If
```

```
26  
27 'On Change of Invoice #, Load Invoice
```

```
28 If Not Intersect(Target, Range("H1" )) Is Nothing And Range("H1" ).Value <> Empty
```

```
29 And Range("H1" ).Value <> Range("B6" ).Value Then Invoice_Load 'Load Invoice
```

```
30 End Sub
```

```
31 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
32 CheckForHelp
```

```
33 End Sub
```

1 Option Explicit

2

1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Workbook_Open()
4
5 End Sub
6
7 Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range,
8     Cancel As Boolean)
9     RightClick_ShowButtons
10 End Sub
11
12 Private Sub Workbook_WindowDeactivate(ByVal Wn As Window)
13     Rightclick_Cancel
14 End Sub
```

1 Option Explicit

2


```
1 Option Explicit
2
3 Private Sub CancelBtn_Click()
4     Unload Me
5 End Sub
6
7
8 Private Sub SaveBtn_Click()
9     SaveHelp 'Run Macro To Save Help
10 End Sub
```

```

1  Option Explicit
2
3  Sub Contact_AddNew()
4      With CustMgr
5          .Range("B12,N4:N12" ).ClearContents 'Clear Existing Contact
6          .Shapes("ExistContGrp" ).Visible = msoFalse
7          .Shapes("SaveContBtn" ).Visible = msoCTrue
8      End With
9  End Sub
10
11 Sub Contact_Delete()
12     Dim ContRow As Long, SelRow As Long
13     With CustMgr
14         If MsgBox("Are you sure you want to delete this contact?" , vbYesNo, "Delete
15         Contact" ) = vbNo Then Exit Sub
16         If .Range("B12" ).Value = Empty Then Exit Sub
17         SelRow = .Range("B12" ).Value 'Contact Row
18         ContRow = .Range("R" & SelRow).Value
19         If ContRow <> Empty Then Contacts.Range(ContRow & ":" & ContRow).EntireRow.
20         Delete
21         LoadContacts
22         If .Range("M16" ).Value <> Empty Then .Range("M16" ).Select Else: Contact_AddNew
23     End With
24 End Sub
25
26 Sub Contact_Load()
27     Dim ContRow As Long, ContCol As Long
28     With CustMgr
29         .Range("B13" ).Value = True 'Set Cont Load To True
30         ContRow = .Range("B12" ).Value 'Contact Row
31
32         For ContCol = 13 To 17
33             .Range(.Cells(30, ContCol).Value).Value = .Cells(ContRow, ContCol).Value
34             'Contact details
35         Next ContCol
36         .Range("B13" ).Value = False 'Set cont load to false
37     End With
38 End Sub
39
40 Sub Contact_SaveNew()
41     Dim ContDbRow As Long, ContDbCol As Long
42     With CustMgr
43         If .Range("N4" ).Value = Empty Then
44             MsgBox "Please enter a Contact First Name"
45             Exit Sub
46         End If
47         ContDbRow = Contacts.Range("A99999" ).End(xlUp).Row + 1 'First Avail Cont Row
48
49         For ContDbCol = 1 To 6
50             Contacts.Cells(ContDbRow, ContDbCol).Value = .Range(Contacts.Cells(1,
51             ContDbCol).Value).Value
52         Next ContDbCol
53         Contacts.Range("G" & ContDbRow).Value = "=Row()"
54         LoadContacts
55         .Range("B12" ).Value = .Range("M29" ).End(xlUp).Row
56         .Shapes("ExistContGrp" ).Visible = msoCTrue
57         .Shapes("SaveContBtn" ).Visible = msoFalse
58     End With
59 End Sub

```

```

1 Option Explicit
2
3 Sub Customer_Load()
4 Dim CustRow As Long, LastInvRow As Long, LastW0Row As Long, CustCol As Long
5 With CustMgr
6   If .Range("B10").Value = Empty Then
7     MsgBox "Please select a correct customer name from the list"
8     Exit Sub
9   End If
10  CustRow = .Range("B10").Value
11
12  'Clear Existing Data
13  .Range("B12,E4:G8,I4,K4,I6:K6,E8:G8,I8:K8,E10:K10,E12:G12,I12,K12,D14:K16").
  ClearContents 'Clear Customer Cells
14  .Range("D20:G29,I20:K29,M16:N29,N4:N12").ClearContents 'Clear All Other Fields
15
16  For CustCol = 1 To 13
17    .Range(Customers.Cells(1, CustCol).Value).Value = Customers.Cells(CustRow,
  CustCol).Value
18  Next CustCol
19  .Shapes("ExistCustGrp").Visible = msoCTrue
20  .Shapes("NewCustGrp").Visible = msoFalse
21  .Shapes("SaveContBtn").Visible = msoFalse
22  .Shapes("ExistContGrp").Visible = msoCTrue
23
24  LoadContacts 'Load Contacts
25
26  'Load Invoice
27  LastInvRow = InvoiceList.Range("A99999").End(xlUp).Row
28  InvoiceList.Range("I3:L99999").ClearContents 'Clear any previous results
29  InvoiceList.Range("A2:F" & LastInvRow).AdvancedFilter xlFilterCopy,
  CriteriaRange:=InvoiceList.Range("H2:H3"), CopyToRange:=InvoiceList.Range(
  "J2:M2"), Unique:=False
30  .Range("D20:G29").Value = InvoiceList.Range("J3:M12").Value 'Bring in all
  Invoice data
31
32  'Load Work Orders
33  LastW0Row = WorkOrders.Range("A99999").End(xlUp).Row
34  WorkOrders.Range("H3:J99999").ClearContents 'Clear any previous results
35  WorkOrders.Range("A2:D" & LastW0Row).AdvancedFilter xlFilterCopy, CriteriaRange:
  =WorkOrders.Range("F2:F3"), CopyToRange:=WorkOrders.Range("H2:J2"), Unique:=
  False
36  .Range("I20:K29").Value = WorkOrders.Range("H3:J12").Value 'Bring in all W0
  Data
37 End With
38 End Sub
39
40
41 Sub LoadContacts()
42 Dim LastContRow As Long
43 LastContRow = Contacts.Range("A99999").End(xlUp).Row
44 Contacts.Range("L3:Q999").ClearContents 'Clear any previous results
45 Contacts.Range("A3:G" & LastContRow).AdvancedFilter xlFilterCopy, Contacts.Range(
  "J2:J3"), CopyToRange:=Contacts.Range("L2:Q2"), Unique:=False
46 CustMgr.Range("M16:R29").Value = Contacts.Range("L3:Q16").Value 'Copy over
  any/all contacts
47 End Sub

```

```
1 Option Explicit
2
3 Sub Customer_AddNew()
4     With CustMgr
5         .Range("B6" ).Value = True 'Set new customer to true
6         .Shapes("ExistCustGrp" ).Visible = msoFalse
7         .Shapes("NewCustGrp" ).Visible = msoCTrue
8         'Clear Existing Data
9         .Range(
10            "B2,F2:G2,B12,E4:G8,I4,K4,I6:K6,E8:G8,I8:K8,E10:K10,E12:G12,I12,K12,D14:K16" ).
11            ClearContents 'Clear Customer Cells
12        End With
13    End Sub
14
15 Sub Customer_CancelNew()
16     If Customers.Range("B4" ).Value <> Empty Then CustMgr.Range("F2" ).Value = Customers
17         .Range("B4" ).Value
18     End Sub
19
20 Sub Customer_Delete()
21     If MsgBox("Are you sure you want to delete this Customer?" , vbYesNo, "Delete
22         Customer" ) = vbNo Then Exit Sub
23     Dim CustRow As Long
24     If CustMgr.Range("B4" ).Value <> Empty Then
25         CustRow = CustMgr.Range("B4" ).Value 'Customer Row
26         Customers.Range(CustRow & ":" & CustRow).EntireRow.Delete
27         Customer_CancelNew
28     End If
29 End Sub
```

```
1 Option Explicit
2
3 Sub Customer_SaveUpdate()
4     Dim CustRow As Long, CustCol As Long
5     With CustMgr
6         If .Range("E4" ).Value = Empty Then
7             MsgBox "Please enter a customer name"
8             Exit Sub
9         End If
10        'Determine if new or existing customer
11        If .Range("B4" ).Value = Empty Then 'New customer
12            CustRow = Customers.Range("A99999" ).End(xlUp).Row + 1 'First Avail Customer
13            .Range("B2" ).Value = .Range("B8" ).Value
14            Customers.Range("A" & CustRow).Value = .Range("B8" ).Value 'Next Cust ID #
15        Else 'Existing customer
16            CustRow = .Range("B4" ).Value 'Customer Row
17        End If
18
19        For CustCol = 2 To 13
20            Customers.Cells(CustRow, CustCol).Value = .Range(Customers.Cells(1, CustCol).
21                Value).Value
22        Next CustCol
23        .Shapes("ExistCustGrp" ).Visible = msoCTrue
24        .Shapes("NewCustGrp" ).Visible = msoFalse
25        .Shapes("SaveContBtn" ).Visible = msoFalse
26        .Shapes("ExistContGrp" ).Visible = msoCTrue
27        .Range("B6" ).Value = False 'Existing customer
28    End With
29 End Sub
```

```

1 Option Explicit
2 #If VBA7 Then
3 Private Declare PtrSafe Function GetDC& Lib "user32.dll" (ByVal hwnd&)
4 Private Declare PtrSafe Function GetDeviceCaps& Lib "gdi32" (ByVal hdc&, ByVal nIndex&
5 )
6 #Else
7 Private Declare Function GetDC& Lib "user32.dll" (ByVal hwnd&)
8 Private Declare Function GetDeviceCaps& Lib "gdi32" (ByVal hdc&, ByVal nIndex&)
9 #End If
10 Dim CellAdd As String, ActSheet As String, SrchString As String, HelpText As String
11 Dim HelpRow As Long, HelpID As Long
12
13 Sub AddEditHelp()
14     Dim XPos As Double, YPos As Double
15     With ActiveSheet
16         HelpRow = 0
17         CellAdd = ActiveCell.Address 'Select Cell Address
18         ActSheet = .CodeName 'Active Sheet Code Name
19         SrchString = ActSheet & CellAdd 'Search String
20         On Error Resume Next
21         HelpRow = HelpData.Range("Help_SrchString").Find(SrchString, , xlValues, xlWhole
22         ).Row 'Find Help Row
23         On Error GoTo 0
24         With HelpFrm
25             If HelpRow <> 0 Then 'Existing Help Item
26                 HelpText = HelpData.Range("D" & HelpRow).Value 'Help Text
27                 .HelpID.Value = HelpData.Range("A" & HelpRow).Value 'Help ID
28                 .HelpText.Value = HelpText
29             End If
30             .SheetName.Value = ActSheet 'Sheet Name
31             .CellAddress.Value = CellAdd 'Cell Address
32             .StartupPosition = 0 'Set Start position to 0 initially
33             ' 1 inch = 72 points for usually 96 or 120 dpi
34             'Form Positioning Code Found here:
35             https://stackoverflow.com/questions/6116929/align-userform-to-cells-in-excel#
36
37             XPos = GetDeviceCaps(GetDC(0), 88) / 72 'Set Leftmost Horizontal position
38             YPos = GetDeviceCaps(GetDC(0), 90) / 72 'Set Upper Vertical Screen Position
39             .Left = (ActiveWindow.PointsToScreenPixelsX(1 * XPos) * 1 / XPos) + ActiveCell
40             .MergeArea.Width 'Set Forms Left Position (First column)
41             .Top = (ActiveWindow.PointsToScreenPixelsY(ActiveCell.Top * YPos) * 1 / YPos)
42             + ActiveCell.Height 'Set Forms Top Position (One row under selected Row)
43             .Show
44         End With
45     End With
46 End Sub
47
48 Sub CheckForHelp()
49     With ActiveSheet
50         HelpRow = 0
51         On Error Resume Next
52         .Shapes("HelpGrp").Visible = msoFalse
53         On Error GoTo 0
54         CellAdd = ActiveCell.Address 'Select Cell Address
55         ActSheet = .CodeName 'Active Sheet Code Name
56         SrchString = ActSheet & CellAdd 'Search String
57         On Error Resume Next
58         HelpRow = HelpData.Range("Help_SrchString").Find(SrchString, , xlValues, xlWhole
59         ).Row 'Find Help Row
60         On Error GoTo 0

```

1 2

```

1 2
55 On Error Resume Next
56 With .Shapes("HelpBtn" )
57     If HelpRow <> 0 Then
58         .Left = ActiveSheet.Range(ActiveCell.MergeArea.Address).Left + ActiveCell.
MergeArea.Width - 12
59         .Top = ActiveSheet.Range(CellAdd).Top + 2
60         .Visible = msoCTrue
61
62     Else 'Not Found
63         .Visible = msoFalse
64     End If
65 End With
66 On Error GoTo 0
67 End With
68 End Sub
69
70 Sub CloseHelp()
71     On Error Resume Next
72     ActiveSheet.Shapes("HelpGrp" ).Visible = msoFalse
73     On Error GoTo 0
74 End Sub
75
76
77 Sub DeleteHelp()
78     With ActiveSheet
79         HelpRow = 0
80         If MsgBox("Are you sure you want to delete this Help Item?" , vbYesNo, "Delete
Help" ) = vbNo Then Exit Sub
81         CellAdd = ActiveCell.Address 'Select Cell Address
82         ActSheet = .CodeName 'Active Sheet Code Name
83         SrchString = ActSheet & CellAdd 'Search String
84         On Error Resume Next
85         HelpRow = HelpData.Range("Help_SrchString" ).Find(SrchString, , xlValues, xlWhole
).Row 'Find Help Row
86         On Error GoTo 0
87         If HelpRow = 0 Then Exit Sub
88         HelpData.Range(HelpRow & ":" & HelpRow).EntireRow.Delete
89         .Shapes("HelpGrp" ).Visible = msoFalse
90         .Shapes("HelpBtn" ).Visible = msoFalse
91     End With
92
93
94
95 End Sub
96
97 Sub Rightclick_Cancel()
98     With Application
99         On Error Resume Next
100        .CommandBars("Cell" ).Controls("Add/Edit Help Item" ).Delete
101        .CommandBars("Cell" ).Controls("Delete Help Item" ).Delete
102        On Error GoTo 0
103    End With
104 End Sub
105
106 Sub RightClick_ShowButtons()
107     Dim EditBtn As CommandBarButton, DelBtn As CommandBarButton
108     On Error Resume Next
109     With Application
110         .CommandBars("Cell" ).Controls("Add/Edit Help Item" ).Delete
111         .CommandBars("Cell" ).Controls("Delete Help Item" ).Delete
112         If [UserType] <> "Admin" Then Exit Sub 'Only create right click function for
1 2

```

```

1 2
'Admin' user types
113 Set EditBtn = .CommandBars("Cell" ).Controls.Add(Before:=1, Type:=
msoControlButton)
114 With EditBtn
115     .Caption = "Add/Edit Help Item"
116     .FaceId = 11181
117     .OnAction = "AddEditHelp"
118 End With
119 Set DelBtn = .CommandBars("Cell" ).Controls.Add(Before:=2, Type:=msoControlButton
)
120 With DelBtn
121     .Caption = "Delete Help Item"
122     .FaceId = 1592
123     .OnAction = "DeleteHelp"
124 End With
125 End With
126 End Sub

127
128 Sub SaveHelp()
129     With HelpFrm
130         If .HelpText.Value = Empty Then
131             MsgBox "Please make sure to add a Help Text before saving"
132             Exit Sub
133         End If
134         'Determine if new/existing help
135         If .HelpID.Value = Empty Then 'new Helo
136             On Error Resume Next
137             HelpID = Application.WorksheetFunction.Max([Help_ID]) + 1 'First Avail Help
ID
138             On Error GoTo 0
139             If HelpID = 0 Then HelpID = 1 'Set 1 when there is no data
140             HelpRow = HelpData.Range("A9999" ).End(xlUp).Row + 1 'First Avail Row
141             HelpData.Range("A" & HelpRow).Value = HelpID 'Set Help ID
142             HelpData.Range("B" & HelpRow).Value = ActiveSheet.CodeName 'Sheet Code Name
143             HelpData.Range("C" & HelpRow).Value = ActiveCell.Address 'Cell Address
144             HelpData.Range("E" & HelpRow).Value = ActiveSheet.CodeName & ActiveCell.
Address 'Search String
145         Else 'Existing Help
146             On Error Resume Next
147             HelpRow = HelpData.Range("Help_ID" ).Find(HelpID, , xlValues, xlWhole).Row
'Find Help Row
148             If HelpRow = 0 Then
149                 MsgBox "Help Not Found"
150                 Exit Sub
151             End If
152             On Error GoTo 0
153         End If
154         HelpData.Range("D" & HelpRow).Value = .HelpText.Value 'Help Text
155         Unload HelpFrm
156     End With
157 End Sub

158
159 Sub ShowHelpPopUp()
160     With ActiveSheet
161         HelpRow = 0
162         CellAdd = ActiveCell.Address 'Select Cell Address
163         ActSheet = .CodeName 'Active Sheet Code Name
164         SrchString = ActSheet & CellAdd 'Search String
165         On Error Resume Next
166         HelpRow = HelpData.Range("Help_SrchString" ).Find(SrchString, , xlValues, xlWhole
).Row 'Find Help Row

```

1 2


```
167 | 1 2 On Error GoTo 0
168 |     If HelpRow = 0 Then Exit Sub
169 |     HelpText = HelpData.Range("D" & HelpRow).Value 'Help Text
170 |     With .Shapes("HelpGrp" )
171 |         .Left = ActiveSheet.Range(ActiveCell.MergeArea.Address).Left + ActiveSheet.
172 |             Range(ActiveCell.MergeArea.Address).Width
173 |         .Top = ActiveSheet.Range(ActiveCell.MergeArea.Address).Top + ActiveSheet.Range
174 |             (ActiveCell.MergeArea.Address).Height
175 |         With .GroupItems("HelpPopUp" )
176 |             .TextFrame2.TextRange.Text = HelpText
177 |             .TextFrame2.AutoSize = msoAutoSizeShapeToFitText
178 |             .Width = 150 'Set Fixed width (height Dynamic based on text)
179 |         End With
180 |     End With
181 | End Sub
```

```

1 Option Explicit
2
3 Sub Invoice_CancelNew()
4     If InvoiceList.Range("A3" ).Value <> Empty Then Invoice.Range("H1" ).Value =
5         InvoiceList.Range("A3" ).Value
6     End Sub
7
8 Sub Invoice_Load()
9     Dim InvRow As Long, InvItemRow As Long, LastInvItemRow As Long
10    Dim ResultRow As Long, LastItemResultRow As Long
11    With Invoice
12        If .Range("B3" ).Value = Empty Then
13            MsgBox "Please select a correct Invoice #"
14            Exit Sub
15        End If
16        InvRow = .Range("B3" ).Value 'Invoice Row
17        .Range("E3,E4,E5,D8:G27,I8:I27" ).ClearContents 'Clear Invoice Fields
18        .Range("B5" ).Value = True 'Set Invoice Load To True
19        .Range("H2" ).Value = InvoiceList.Range("B" & InvRow).Value 'Date
20        .Range("E3" ).Value = InvoiceList.Range("D" & InvRow).Value 'Customer
21        .Range("H3" ).Value = InvoiceList.Range("E" & InvRow).Value 'Status
22        'Load Invoice Items
23        LastInvItemRow = InvoiceItems.Range("A9999" ).End(xlUp).Row 'Last Item Row
24        If LastInvItemRow < 3 Then GoTo NoItems
25        InvoiceItems.Range("A2:G" & LastInvItemRow).AdvancedFilter xlFilterCopy,
26        CriteriaRange:=InvoiceItems.Range("I2:I3" ), CopyToRange:=InvoiceItems.Range(
27        "K2:Q2" ), Unique:=False
28        LastItemResultRow = InvoiceItems.Range("K9999" ).End(xlUp).Row
29        If LastItemResultRow < 3 Then GoTo NoItems
30        For ResultRow = 3 To LastItemResultRow
31            InvItemRow = InvoiceItems.Range("P" & ResultRow).Value 'Invoice Row
32            .Range("D" & InvItemRow & ":G" & InvItemRow).Value = InvoiceItems.Range("L"
33            & ResultRow & ":O" & ResultRow).Value 'Item, Desc, Qty, Price
34            .Range("I" & InvItemRow).Value = InvoiceItems.Range("Q" & ResultRow).Value
35            'Bring over item row
36        Next ResultRow
37        NoItems:
38        .Range("B4" ).Value = False 'Set new invoice to false
39        .Range("B5" ).Value = False 'Set Invoice Load To False
40        .Shapes("CancelNewBtn" ).Visible = msoFalse
41        .Shapes("AddNewBtn" ).Visible = msoTrue
42    End With
43 End Sub
44
45 Sub Invoice_New()
46     With Invoice
47         .Range("B4" ).Value = True 'Set new Invoice to true
48         .Range("H1,E3,H3,E4,E5,D8:G27,I8:I27" ).ClearContents 'Clear Invoice Fields
49         .Range("H2" ).Value = Now
50         .Shapes("CancelNewBtn" ).Visible = msoTrue
51         .Shapes("AddNewBtn" ).Visible = msoFalse
52         .Range("H1" ).Value = .Range("B6" ).Value 'Add New Invoice #
53     End With
54 End Sub
55
56 Sub Invoice_Print()
57     Invoice.PrintOut , , , False, , , , False
58 End Sub
59
60 Sub Invoice_SaveUpdate()
61     Dim InvRow As Long, InvItemRow As Long, LastItemRow As Long, ItemRow As Long

```

1

```

1
57 With Invoice
58 'Determine New or Existing Invoice
59 If .Range("B3").Value = Empty Then 'New Invoice
60     InvRow = InvoiceList.Range("A9999").End(xlUp).Row + 1 'First Available Row
61     .Range("H1").Value = .Range("B6").Value 'Add Invoice Number
62     InvoiceList.Range("A" & InvRow).Value = .Range("B6").Value 'Add Invoice #
63 Else: 'Existing Invoice
64     InvRow = .Range("B3").Value 'Existing Invoice Row
65 End If
66 InvoiceList.Range("B" & InvRow).Value = .Range("H2").Value 'Date
67 InvoiceList.Range("C" & InvRow).Value = .Range("B1").Value 'Customer ID
68 InvoiceList.Range("D" & InvRow).Value = .Range("E3").Value 'Customer Name
69 InvoiceList.Range("E" & InvRow).Value = .Range("H3").Value 'Status
70 InvoiceList.Range("F" & InvRow).Value = .Range("H30").Value 'Invoice Total
71
72 'Add / Update Invoice Items
73 LastItemRow = .Range("D27").End(xlUp).Row 'Last Invoice Item Row
74 If LastItemRow < 8 Then GoTo NoItems
75 For ItemRow = 8 To LastItemRow
76     If .Range("I" & ItemRow).Value <> Empty Then 'existing Row
77         InvItemRow = .Range("I" & ItemRow).Value
78     Else 'New Row
79         InvItemRow = InvoiceItems.Range("A9999").End(xlUp).Row + 1 'First
80         AvailRow
81         InvoiceItems.Range("A" & InvItemRow).Value = .Range("H1").Value
82         'Invoice #
83         .Range("I" & ItemRow).Value = InvItemRow 'Invoice Item Row
84         InvoiceItems.Range("G" & InvItemRow).Value = "=Row()" 'Add Row #
85         InvoiceItems.Range("B" & InvItemRow & ":E" & InvItemRow).Value = .Range("D"
86         & ItemRow & ":G" & ItemRow).Value 'Add Desc, Qty, Price
87         InvoiceItems.Range("F" & InvItemRow).Value = ItemRow
88     Next ItemRow
89 NoItems:
90     .Range("B4").Value = False 'Set new invoice to false
91     .Shapes("CancelNewBtn").Visible = msoFalse
92     .Shapes("AddNewBtn").Visible = msoCTrue
93 End With
94 End Sub

```

—
→, 7

A

ActiveCell, 18-21
ActiveSheet, 18-21
ActiveWindow, 18
ActSheet, 18-20
Add, 20
AddEditHelp, 18
Address, 18-21
AdvancedFilter, 15, 22
Application, 19, 20
AutoSize, 21

B

Before, 20

C

Cancel, 11
CancelBtn_Click, 13
Caption, 20
CellAdd, 18-20
CellAddress, 18
Cells, 4, 14, 15, 17
CheckForHelp, 4, 7, 18
ClearContents, 14-16, 22
CloseHelp, 19
CodeName, 18-20
CommandBarButton, 19
CommandBars, 19, 20
Contact_AddNew, 14
Contact_Delete, 14
Contact_Load, 4, 14
Contact_SaveNew, 14
Contacts, 4, 14, 15
ContCol, 14
ContDbCol, 14
ContDbRow, 4, 14
Controls, 19, 20
ContRow, 14
CopyToRange, 15, 22
CriteriaRange, 15, 22
CustCol, 15, 17
CustMgr, 14-17
Customer_AddNew, 16
Customer_CancelNew, 16
Customer_Delete, 16
Customer_Load, 4, 15
Customer_SaveUpdate, 17
Customers, 7, 15-17
CustRow, 7, 15-17

D

DelBtn, 19, 20
Delete, 14, 16, 19
DeleteHelp, 19

E

EditBtn, 19, 20
Empty, 4, 7, 14-17, 20, 22, 23
EntireRow, 14, 16, 19
Explicit, 3-18, 22

F

Faceld, 20
Find, 18-20

G

GetDC, 18
GetDeviceCaps, 18
GroupItems, 21

H

hDC, 18
Height, 18, 21
Help_ID, 20
HelpData, 18-21
HelpFrm, 18, 20
HelpID, 18, 20
HelpRow, 18-21
HelpText, 18, 20, 21
hwnd, 18

I

Intersect, 4, 7
InvItemRow, 22, 23
Invoice, 22, 23
Invoice_CancelNew, 22
Invoice_Load, 7, 22
Invoice_New, 22
Invoice_Print, 22
Invoice_SaveUpdate, 22
InvoiceItems, 22, 23
InvoiceList, 15, 22, 23
InvRow, 22, 23
ItemRow, 7, 22, 23
Items, 7

L

LastContRow, 15
LastInvItemRow, 22
LastInvRow, 15
LastItemResultRow, 22
LastItemRow, 22, 23
LastWORow, 15
Left, 18, 19, 21
LoadContacts, 4, 14, 15

M

Max, 20
MergeArea, 18, 19, 21
MsgBox, 14-17, 19, 20, 22
msoAutoSizeShapeToFitText, 21
msoControlButton, 20
msoCTrue, 14-17, 19, 21-23
msoFalse, 14-19, 22, 23

N

nIndex, 18
NoItems, 22, 23
Now, 22

O

Offset, 4
OnAction, 20

P

PointsToScreenPixelsX, 18
PointsToScreenPixelsY, 18
PrintOut, 22
PtrSafe, 18

R

Range, 4, 7, 11, 14-23
ResultRow, 22
Rightclick_Cancel, 11, 19

RightClick_ShowButtons, 11, 19
Row, 4, 7, 14, 15, 17-20, 22, 23

S

SaveBtn_Click, 13
SaveHelp, 13, 20
SelRow, 14
Sh, 11
Shapes, 14-19, 21-23
SheetName, 18
Show, 18
ShowHelpPopUp, 20
SrchString, 18-20
StartupPosition, 18

T

Target, 4, 7, 11
Text, 21
TextFrame2, 21
TextRange, 21
Top, 18, 19, 21

U

Unique, 15, 22
Unload, 13, 20
UserType, 19

V

Value, 4, 7, 14-18, 20-23
VBA7, 18
vbNo, 14, 16, 19
vbYesNo, 14, 16, 19
Visible, 14-19, 21-23

W

Width, 18, 19, 21
Window, 11
Wn, 11
Workbook_Open, 11
Workbook_SheetBeforeRightClick, 11
Workbook_WindowDeactivate, 11
WorkOrders, 15
Worksheet_Change, 4, 7
Worksheet_SelectionChange, 4, 7
WorksheetFunction, 20

X

xlFilterCopy, 15, 22
xlUp, 14, 15, 17, 20, 22, 23
xlValues, 18-20
xlWhole, 18-20
XPos, 18

Y

YPos, 18

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