

# VBA SOURCE CODE BOOK

## THE ULTIMATE EXCEL PROTECTION - UPDATED

EXCLUSIVE FOR  
SILVER & GOLD  
MEMBERS

FEATURE, FIX, OR  
FOCUS



*Excel For Freelancers*

Learn How To Protect Your  
Work With the Ultimate Excel  
Protection Masterclass



[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



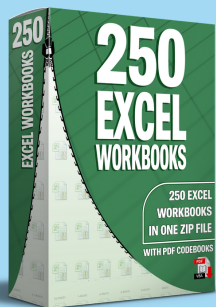
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[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](#)



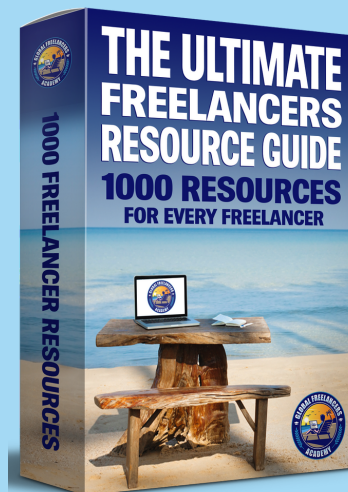
With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[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.....	3
VBAProject.....	3
Documents.....	3
Admin.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub ].....	3
CalPopUp.....	4
(Declarations).....	4
Customers.....	5
(Declarations).....	5
Invoice.....	6
(Declarations).....	6
Worksheet_Change [Sub ].....	6
Worksheet_SelectionChange [Sub ].....	7
InvoiceItems.....	11
(Declarations).....	11
InvoiceList.....	12
(Declarations).....	12
Items.....	13
(Declarations).....	13
Protection.....	14
(Declarations).....	14
Worksheet_SelectionChange [Sub ].....	14
Start.....	15
(Declarations).....	15
ThisWorkbook.....	16
(Declarations).....	16
Workbook_Activate [Sub ].....	16
Workbook_BeforeClose [Sub ].....	16
Workbook_BeforeSave [Sub ].....	16
Workbook_Deactivate [Sub ].....	16
Workbook_Open [Sub ].....	16
Workbook_SheetDeactivate [Sub ].....	16
Users.....	17
(Declarations).....	17
Forms.....	18
AddCustForm.....	18
(Declarations).....	18
CancelBtn_Click [Sub ].....	18
SaveBtn_Click [Sub ].....	18
AddItemForm.....	19
(Declarations).....	19
CancelBtn_Click [Sub ].....	19
SaveBtn_Click [Sub ].....	19
LoginForm.....	20
(Declarations).....	20
CancelBtn_Click [Sub ].....	20
OkBtn_Click [Sub ].....	20
Password_Exit [Sub ].....	20
Username_Exit [Sub ].....	20
Modules.....	21
Admin_Menu_Macros.....	21
(Declarations).....	21
AppEvents_Start [Sub ].....	21
AppEvents_Stop [Sub ].....	21
Company_AddLogo [Sub ].....	21
EditMode [Sub ].....	21
CutCopyPaste_Macros.....	22
(Declarations).....	22
CutCopyPasteDisabled [Sub ].....	22
EnableMenuItem [Sub ].....	22
rxIRibbonUI_onLoad [Sub ].....	22
rxshared_getEnabled [Sub ].....	22
tests [Sub ].....	22
ToggleCutCopyAndPaste [Sub ].....	22
Invoice_Macros.....	24

## Table of Contents

(Declarations) .....	24
Customer_AddUpdate [Sub ] .....	24
Invoice_Delete [Sub ] .....	24
Invoice_Load [Sub ] .....	24
Invoice_New [Sub ] .....	25
Invoice_NextInvoice [Sub ] .....	25
Invoice_PrevInvoice [Sub ] .....	26
Invoice_SavedMsg [Sub ] .....	26
Invoice_SaveUpdate [Sub ] .....	26
Page_Macros .....	28
(Declarations) .....	28
Invoice_DeletePage [Sub ] .....	28
Invoice_NewPage [Sub ] .....	29
Invoice_NextPage [Sub ] .....	29
Invoice_PageLoad [Sub ] .....	29
Invoice_PrevPage [Sub ] .....	29
Page_DeleteLineItem [Sub ] .....	29
PopUp_Calendar .....	31
(Declarations) .....	31
CalCol [Sub ] .....	31
CalendarHide [Sub ] .....	31
CalendarShow [Sub ] .....	31
CalFormulaReplacement [Sub ] .....	32
CheckForSheet [Sub ] .....	32
CreateCalSht [Sub ] .....	33
GroupCal [Sub ] .....	35
MacroLinkRemover [Sub ] .....	35
NextMonth [Sub ] .....	36
NextYear [Sub ] .....	36
PrevMonth [Sub ] .....	36
PrevYear [Sub ] .....	37
ReplaceCalendar [Sub ] .....	37
SelectDay [Sub ] .....	37
ShowSettings [Sub ] .....	37
UnGroupCal [Sub ] .....	37
Print_Macros .....	39
(Declarations) .....	39
Invoice_Print [Sub ] .....	39
Invoice_SaveAsPDF [Sub ] .....	39
Protection_Macros .....	40
(Declarations) .....	40
DisableDragAndDrop [Sub ] .....	40
HideNamedRange [Sub ] .....	40
HideSheet_Hidden [Sub ] .....	40
HideSheet_VeryHidden [Sub ] .....	40
ListAllSheets [Sub ] .....	40
LoopThroughNamedRanges [Sub ] .....	40
Protection_WorksheetDetails [Sub ] .....	40
ProtectWorksheet [Sub ] .....	41
ShowSheet [Sub ] .....	41
ToggleWorksheetTabs [Sub ] .....	41
ToogleWorksheetTabs [Sub ] .....	41
User_MachineCheck [Sub ] .....	41
UserLogin [Sub ] .....	42
UserLogout [Sub ] .....	42

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5
6     'On Default Invoice Status Selection
7     If Not Intersect(Target, Range("F6:F11" )) Is Nothing And Range("E" & Target.Row).
8         Value <> Empty Then
9         Range("F6:F11" ).ClearContents
10        Target.Value = Chr(252)
11        Range("E6" ).Select
12    End If
13
14    'On Default Payment Terms Selection
15    If Not Intersect(Target, Range("J6:J11" )) Is Nothing And Range("H" & Target.Row).
16        Value <> Empty Then
17        Range("J6:J11" ).ClearContents
18        Target.Value = Chr(252)
19        Range("H6" ).Select
20    End If
21 End Sub
```

1 Option Explicit

2



1 Option Explicit

2

```

1 Option Explicit
2 Dim ItemTypeRow As Long, ItemCol As Long, ItemRow As Long, TermRow As Long
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim ItemDBRow As Long, InvItemRow As Long, SelRow As Long
5     If Target.CountLarge > 1 Then Exit Sub
6
7     'On Invoice Search
8     If Not Intersect(Target, Range("F2" )) Is Nothing And Range("F2" ).Value <> Empty
9         Then
10            If Range("B2" ).Value = Empty Then 'Incorrect Invoice #
11                MsgBox "Please select a correct Invoice #"
12                Exit Sub
13            End If
14            Range("B4" ).Value = Range("F2" ).Value 'Set Invoice #
15            Range("F2" ).ClearContents
16            Invoice_Load
17        End If
18
19        'On Item Name Change Load Item Details (But Not On Page/Invoice Load)
20        If Not Intersect(Target, Range("K10:K44" )) Is Nothing And Range("B1" ).Value =
21            False Then
22                'Check For Invoice Save
23                If Range("B5" ).Value = Empty Then 'Invoice Not Save
24                    MsgBox "Please make sure to save the Invoice before adding items"
25                    Application.EnableEvents = False
26                    Target.ClearContents
27                    Application.EnableEvents = True
28                    Exit Sub
29                End If
30                If Target.Value <> Empty Then 'Non Empty
31                    On Error Resume Next
32                    ItemDBRow = Items.Range("Item_Name" ).Find(Target.Value, , xlValues, xlWhole).
33                    Row
34                    On Error GoTo 0
35                    If ItemDBRow = 0 Then
36                        If MsgBox("This item does not exist. Would you like to add it?" , vbYesNo,
37                            "Add New Item?" ) = vbNo Then Exit Sub
38                        AddItemForm.Field1.Value = Target.Value 'Set Default Item Name
39                        AddItemForm.Show
40                    End If
41                End If
42                Range("L" & Target.Row).Value = Items.Range("B" & ItemDBRow).Value
43                'Description
44                Range("M" & Target.Row).Value = Items.Range("C" & ItemDBRow).Value 'Default
45                Qty
46                Range("N" & Target.Row).Value = Items.Range("D" & ItemDBRow).Value
47                'Amount/Price
48                Range("P" & Target.Row).Value = Items.Range("E" & ItemDBRow).Value 'Taxable
49            Else 'Empty Clear Row
50                Range("L" & Target.Row & ":N" & Target.Row).ClearContents 'Clear Item
51                Details
52                Range("P" & Target.Row).ClearContents 'Clear Tax
53            End If
54        End If
55
56        'On Any Invoice Item Detail Change, Save To Database (not on Invoice Load)
57        If Not Intersect(Target, Range("J10:P44" )) Is Nothing And Range("B1" ).Value =
58            False Then

```

```

1 2
51 SelRow = Target.Row 'Set Selected Row
52 'Update Invoice Items DB
53 If Range("Q" & SelRow).Value = Empty Then 'New Invoice Item
54   InvItemRow = InvoiceItems.Range("A9999").End(xlUp).Row + 1 'First Avail.
55   InvoiceItems.Range("A" & InvItemRow).Value = Range("B4").Value 'Invoice
56   InvoiceItems.Range("B" & InvItemRow).Value = Range("B11").Value 'Page
57   InvoiceItems.Range("J" & InvItemRow).Value = SelRow 'Set Invoice Row
58   InvoiceItems.Range("K" & InvItemRow).Value = "=row()" 'Set Row Formula
59   Range("Q" & Target.Row).Value = InvItemRow 'Set Item Row
60 Else 'Existing Invoice Item
61   InvItemRow = Range("Q" & Target.Row).Value 'Existing Item Row
62 End If
63 InvoiceItems.Cells(InvItemRow, Target.Column - 7).Value = Target.Value
64 InvoiceItems.Range("H" & InvItemRow).Value = Range("O" & SelRow).Value 'Update
65 End If
66
67 'On Any Terms Or Inv. Date Change, Update Due Date (not on Invoice Load) (if
68 'applicable)
69 If Not Intersect(Target, Range("E4,E6")) Is Nothing And Range("E4").Value <> Empty
70   And Range("E6").Value <> Empty And Range("B1").Value = False Then
71     On Error Resume Next
72     TermRow = Admin.Range("Terms").Find(Range("E6").Value, , xlValues, xlWhole).Row
73     On Error GoTo 0
74     If TermRow = 0 Then Exit Sub 'Exit if not found
75     Range("E8").Value = Range("E4").Value + Admin.Range("I" & TermRow).Value
76     'Update Due Date
77 End If
78
79 '''Customize Invoice Changes
80 If Range("B10").Value = False Then Exit Sub 'Exit unless in Edit mode
81 Dim ItemLblShp As Shape
82 'On Change of Show or Label
83 If Not Intersect(Target, Range("F13:G22")) Is Nothing And Range("B8").Value =
84   False And Range("E" & Target.Row).Value <> Empty Then
85     ItemTypeRow = Range("B9").Value 'Selected Item Type Row
86     ItemCol = 22 + ((ItemTypeRow - 13) * 4) + Target.Column 'Column to Save To
87     Cells(Target.Row, ItemCol).Value = Target.Value 'Update Data table
88     'On Change of Label, Show/Hide Label (but not on Item Detail Columns)
89     If Not Intersect(Target, Range("G13:G22")) Is Nothing And Range("B9").Value <>
90       16 Then
91       On Error Resume Next
92       Set ItemLblShp = Shapes(Range("E" & Target.Row).Value & "_Lb1")
93       If Target.Value = Empty Then ItemLblShp.Visible = msoFalse Else ItemLblShp.
94       Visible = msoCTrue 'Hide/Show Label based on Label Value
95     End If
96 End If
97 End Sub
98
99 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
100   Dim ItemNm As String
101   Dim ItemShp As Shape, ItemLblShp As Shape, GrpShp As Shape, Cal As Shape
102   If Shapes("DelItemGrp").Visible = True Then Shapes("DelItemGrp").Visible =
103     msoFalse

```

```

1
99  If Target.CountLarge > 2 Then Exit Sub
100
101  On Error Resume Next
102  Set Cal = Shapes("Calendar" )
103  On Error GoTo 0
104  If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
105  If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
106
107  If Not Intersect(Target, Range("E4,E8,J10:J44" )) Is Nothing Then 'Change this to
any cell (or cells) you would like to have the Pop-Up Calendar Appear
108      CheckForSheet
109      CalendarShow
110  Else:
111      CheckForSheet
112      CalendarHide
113  End If
114
115  'On Tax Selection Change but not on Invoice Load
116  On Error Resume Next
117  If Not Intersect(Target, Range("P10:P44" )) Is Nothing And Range("B1" ).Value =
False Then
118      If Target.Value = Chr(252) Then Target.ClearContents Else Target.Value = Chr(252)
119      Range("P8" ).Select 'Select another cell
120  End If
121
122  'On Existing Invoice Item Selection
123  If Not Intersect(Target, Range("J10:P55" )) Is Nothing Then
124      Range("B7" ).Value = Target.Row
125      If Range("K" & Target.Row).Value <> Empty Then
126          With Shapes("DelItemGrp" )
127              .Left = Range("Q" & Target.Row).Left
128              .Top = Range("Q" & Target.Row).Top
129              .Visible = msoCTrue
130          End With
131      End If
132  End If
133
134
135  '''Customize Invoice Changes
136
137  If Range("B10" ).Value = False Then Exit Sub 'Exit unless in Edit mode
138  'On Edit Invoice Item Menu Select
139  If Not Intersect(Target, Range("D13:D16" )) Is Nothing Then
140      Range("B8" ).Value = True 'Set Menu Change to True
141      Range("B9" ).Value = Target.Row 'Set Type /Trigger CF
142      Range("E13:G22" ).ClearContents 'Clear Fields
143      ItemCol = 27 + ((Target.Row - 13) * 4)
144      Range("E13:G23" ).Value = Range(Cells(13, ItemCol), Cells(23, ItemCol + 2)).Value
'Load Item Details
145      Range("B8" ).Value = False 'Set Menu Change To False
146  End If
147
148  'On Show/Hide Selection
149  If Not Intersect(Target, Range("F13:F23" )) Is Nothing And Range("E" & Target.Row).
Value <> Empty Then
150      Dim InvWidth As Double, DescWidth As Double
151      Set ItemShp = Nothing
152      Set ItemLblShp = Nothing

```

1 2

```

1 2
153 If Target.Value = Chr(252) Then Target.ClearContents Else Target.Value = Chr(252)
154 If Target.Row = 23 Then 'Footer Group
155     If Range("F" & Target.Row).Value = Empty Then Shapes("FooterGrp").Visible =
156         msoFalse Else Shapes("FooterGrp").Visible = msoCTrue
157     Range("F2").Select
158     Exit Sub
159 End If
159 ItemTypeRow = Range("B9").Value 'Set Item Type Row
160 ItemRow = Target.Row
161 'If Item Details are selected, then Hide/Show Columns
162 If ItemTypeRow = 16 Then 'Item Details (Columns)
163     ItemCol = Target.Row - 3 'Set Item Column
164     If Target.Value = "" Then 'Hide Column
165         Cells(1, ItemCol).EntireColumn.Hidden = True
166     Else 'Show Column
167         Cells(1, ItemCol).EntireColumn.Hidden = False
168     End If
169     'Adjust Item Description column if its not hidden
170     If ItemCol <> 12 Then
171         InvWidth = Columns("J").ColumnWidth + Columns("K").ColumnWidth + Columns(
172             "L").ColumnWidth + Columns("M").ColumnWidth +
173             Columns("N").ColumnWidth + Columns("O").ColumnWidth + Columns("P").
174             ColumnWidth
175             DescWidth = Columns("L").ColumnWidth
176             If InvWidth > 546.5 Then Columns("L").ColumnWidth = (DescWidth - (InvWidth
177                 - 93.6))
178             If InvWidth < 546.5 Then Columns("L").ColumnWidth = (DescWidth + (93.6 -
179                 InvWidth))
180     End If
181 Else 'Invoice Fields (Shapes)
182     ItemCol = 29 + ((ItemTypeRow - 13) * 4)
183     ItemNm = Range("E" & Target.Row).Value 'Set Item Name
184     'Check For Item Value Shape
185     On Error Resume Next
186     Set ItemShp = Shapes(ItemNm)
187     On Error GoTo 0
188     If ItemShp Is Nothing Then 'Create & Position Value Shape if it does not
189     exist
190     If ItemNm <> "Logo" Then
191         Shapes("SampleInvShp").Duplicate.Name = ItemNm 'Create Shape If
192         missing
193     Else 'Logo
194         If Admin.Range("C10").Value = Empty Or Dir(Admin.Range("C10").Value,
195             vbDirectory) = "" Then GoTo CheckForLbl
196         On Error Resume Next
197         Pictures.Insert(Admin.Range("C10").Value).Name = ItemNm 'Set Logo
198         Picture
199         On Error GoTo 0
200     End If
201     Set ItemShp = Shapes(ItemNm)
202     With ItemShp
203         If ItemNm <> "Logo" Then .DrawingObject.Formula = "=" & Cells(ItemRow,
204             ItemCol + 1).Address
205         .Left = Range("L1").Left
206         .Top = Range("L5").Top
207         .Placement = xlMove 'Move but not size with cells (columns)
208     End With

```

```

202     1 2 3 4 End If
203     CheckForLbl:
204     'Check For Item Label Shape
205     On Error Resume Next
206     Set ItemLblShp = Shapes(ItemNm & "_Lbl" )
207     On Error GoTo 0
208     If ItemLblShp Is Nothing Then 'Create & Position Value Shape if it does not exist
209         Shapes("SampleInvShp" ).Duplicate.Name = ItemNm & "_Lbl" 'Create Shape If missing
210         Set ItemLblShp = Shapes(ItemNm & "_Lbl" )
211         With ItemLblShp
212             .DrawingObject.Formula = "=" & Cells(ItemRow, ItemCol).Address
213             .Left = Range("L1" ).Left - 50
214             .Top = Range("L5" ).Top
215             .Placement = xlMove
216         End With
217     End If
218
219     'Group Shapes if group does not exist
220     On Error Resume Next
221     Set GrpShp = Shapes(ItemNm & "_Grp" )
222     On Error GoTo 0
223     If GrpShp Is Nothing Then 'Group Shapes
224         Shapes.Range(Array(ItemShp.Name, ItemLblShp.Name)).Group.Name = ItemNm &
225             "_Grp"
226         Shapes(ItemNm & "_Grp" ).Placement = xlMove 'Set to Move but not size with cells
227     End If
228     If Target.Value = Empty Or Target.Offset(0, 1).Value = Empty Then 'Ungroup and Hide on Empty Label
229         On Error Resume Next
230         Shapes(ItemNm & "_Grp" ).Ungroup
231         On Error Resume Next
232         ItemLblShp.Visible = msoFalse
233     Else
234         ItemLblShp.Visible = msoCTrue 'Show/Hide Invoice Label
235     End If
236     If Target.Value = Empty Then 'Hidden
237         ItemShp.Visible = msoFalse
238     Else 'Visible
239         ItemShp.Visible = msoCTrue 'Show/Hide Invoice Field Value
240     End If
241 End If
242     Range("F2" ).Select
243 End If
244 End Sub

```

---

1	Option Explicit
2	

1 Option Explicit

2



1

Option Explicit

```
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("O5:Q50" )) Is Nothing And Range("N" & Target.Row).
6         Value <> Empty Then
7         Select Case Target.Value
8             Case Is = "" , "x"
9                 Target.Value = Chr(208) 'Unprotected & Visible
10                Case Is = Chr(208) 'Currently Unprotected & visbile
11                Target.Value = Chr(207) 'Protected & visible
12            Case Else
13                Target.Value = Chr(120) 'Protected & Hidden
14            End Select
15            Range("S5" ).Select 'Select Outside Grid
16        End If
17    End Sub
```

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Workbook_Activate()
4     Call ToggleCutCopyAndPaste(True)
5 End Sub
6
7 Private Sub Workbook_BeforeClose(Cancel As Boolean)
8     'UserLogout
9     'Call ToggleCutCopyAndPaste(True)
10 End Sub
11
12 Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
13     'If SaveAsUI Then
14     '    MsgBox "The 'Save As' function has been disabled.", vbInformation, "Save As
15     '    Disabled"
16     '    Cancel = True
17     'End If
18 End Sub
19
20 Private Sub Workbook_Deactivate()
21     Call ToggleCutCopyAndPaste(True)
22 End Sub
23
24 Private Sub Workbook_Open()
25     Call ToggleCutCopyAndPaste(True)
26     ToggleWorksheetTabs (True)
27     ProtectWorksheet
28     ' Protection_WorksheetDetails 'Protect Worksheet based on user type
29 End Sub
30
31 Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)
32     ' On Error Resume Next
33     ' grxIRibbonUI_CutCopyPaste.Invalidate
34     ' On Error GoTo 0
35 End Sub
```

---

1	Option Explicit
2	

```
1 Option Explicit
2
3 Private Sub CancelBtn_Click()
4     AddCustForm.Hide
5 End Sub
6
7 Private Sub SaveBtn_Click()
8     Dim CustomerFld As Control
9     Dim CustRow As Long, CustCol As Long
10    If Me.Field1.Value = Empty Then
11        MsgBox "Please make sure to save a Customer Name before saving"
12        Exit Sub
13    End If
14    With Customers
15        If Invoice.Range("B3").Value = Empty Then 'New Customer
16            CustRow = .Range("A99999").End(xlUp).Row + 1 'First avail Row
17            On Error Resume Next
18            .Range("A" & CustRow).Value = Application.WorksheetFunction.Max(.Range(
19                "Cust_ID" )) + 1
20            On Error GoTo 0
21            If .Range("A" & CustRow).Value = Empty Then .Range("A" & CustRow).Value = 1
22                'Set First Cust ID to 1
23        Else
24            CustRow = Invoice.Range("B3").Value
25        End If
26        For CustCol = 2 To 8
27            Set CustomerFld = Me.Controls("Field" & CustCol - 1)
28            .Cells(CustRow, CustCol).Value = CustomerFld.Value
29        Next CustCol
30        AddCustForm.Hide
31        Invoice.Range("E5").Value = Field1.Value 'Set Customer Name
32    End With
33 End Sub
```

```
1 Option Explicit
2
3
4 Private Sub CancelBtn_Click()
5     AddItemForm.Hide
6 End Sub
7
8 Private Sub SaveBtn_Click()
9     Dim SelRow As Long
10    Dim ItemFld As Control
11    Dim ItemRow As Long, ItemCol As Long
12    SelRow = Invoice.Range("B7").Value 'Selected Row
13    If Me.Field1.Value = Empty Then
14        MsgBox "Please make sure to add an Item Name before saving"
15        Exit Sub
16    End If
17    With Items
18        ItemRow = .Range("A99999").End(xlUp).Row + 1 'First avail Row
19        On Error Resume Next
20        For ItemCol = 1 To 4
21            Set ItemFld = Me.Controls("Field" & ItemCol)
22            .Cells(ItemRow, ItemCol).Value = ItemFld.Value
23        Next ItemCol
24        If Me.TaxField.Value = True Then .Range("E" & ItemRow).Value = Chr(252) 'Set
25        Taxable Checkbox
26        AddItemForm.Hide
27        Invoice.Range("K" & SelRow).Value = Field1.Value 'Set Item Name
28    End With
29 End Sub
```

```
1
2 Option Explicit
3
4 Private Sub CancelBtn_Click()
5     LoginForm.Hide
6     Username.Text = ""
7     Password.Text = ""
8     Users.Range("B1,B2" ).ClearContents
9 End Sub
10
11 Private Sub OkBtn_Click()
12     Dim WkSht As Worksheet
13     If Users.Range("B3" ).Value = True Then 'Correct Password
14         Users.Range("B4" ).Value = Users.Range("B1" ).Value 'Set Current User
15         LoginForm.Hide
16         For Each WkSht In ThisWorkbook.Worksheets
17             If WkSht.Name = "Admin" Then
18                 If Users.Range("B5" ).Value = "True" Then 'Admin
19                     WkSht.Visible = xlSheetVisible
20                 Else: 'Non-Admin
21                     WkSht.Visible = xlSheetVeryHidden
22                 End If
23             Else: ' Not Admin
24                 WkSht.Visible = xlSheetVisible 'Non Admin Sheets
25             End If
26         Next WkSht
27         Users.Range("B1,B2" ).ClearContents
28         If Invoice.Range("B10" ).Value = True Then EditMode 'Switch off Edit Mode for
29             Invoicing if it is on
30             Invoice.Activate 'Set Invoice as Active Sheet
31     Else:
32         MsgBox "Please enter a correct Username & Password"
33     End If
34 End Sub
35
36 Private Sub Password_Exit(ByVal Cancel As MSForms.ReturnBoolean)
37     Users.Range("B2" ).Value = Me.Password.Value
38 End Sub
39
40
41 Private Sub Username_Exit(ByVal Cancel As MSForms.ReturnBoolean)
42     Users.Range("B1" ).Value = Me.Username.Value
43 End Sub
```



```

1  Option Explicit
2
3  Sub AppEvents_Start()
4      Application.EnableEvents = True
5      Application.Calculation = xlCalculationAutomatic
6      Application.ScreenUpdating = True
7  End Sub
8
9
10 Sub AppEvents_Stop()
11     Application.EnableEvents = False
12     Application.Calculation = xlCalculationManual
13     Application.ScreenUpdating = False
14 End Sub
15
16 Sub Company_AddLogo()
17     Dim LogoPic As FileDialog, PicPath As String
18     Set LogoPic = Application.FileDialog(msoFileDialogFilePicker)
19     With LogoPic
20         .Title = "Please select a logo picture"
21         .Filters.Add "Picture Files" , "*.jpg,*.png,*.gif" , 1
22         .AllowMultiSelect = False
23         If .Show <> -1 Then GoTo NoSelection
24         PicPath = .SelectedItems(1)
25     End With
26     Admin.Range("C10" ).Value = PicPath
27     NoSelection:
28 End Sub
29
30
31 Sub EditMode()
32     Dim MovInc As Long, BtnMov As Long
33     If Users.Range("B5" ).Value = False Then 'Non Admin User
34         MsgBox "You do not have admin right to make changes to this Invoice. Please
35             check with your administrator"
36         Exit Sub
37     End If
38     With Invoice
39         If .Shapes("EditModeBack" ).TextFrame2.TextRange.Text = "Off" Then 'Turn On
40             .Shapes("EditModeSwitch" ).Fill.ForeColor.RGB = RGB(0, 176, 80) ' Turn To
41             Green Color
42             .Shapes("EditModeBack" ).TextFrame2.TextRange.Text = "On"
43             MovInc = 1
44             .Range("B10" ).Value = True 'Set Edit Mode To True
45         Else 'Turn Off
46             MovInc = -1
47             .Shapes("EditModeSwitch" ).Fill.ForeColor.RGB = RGB(127, 125, 127) 'Turn to
48             gray color
49             .Shapes("EditModeBack" ).TextFrame2.TextRange.Text = "Off"
50             .Range("B10" ).Value = False 'Set Edit Mode To False
51         End If
52     End With
53     Application.ScreenUpdating = False
54     For BtnMov = 1 To 29
55         .Shapes("EditModeSwitch" ).Left = .Shapes("EditModeSwitch" ).Left + MovInc
56     Next BtnMov
57     Application.ScreenUpdating = True
58 End Sub

```

```
1
2 Option Explicit
3
4 Public grxIRibbonUI_CutCopyPaste As IRibbonUI
5
6 Sub CutCopyPasteDisabled()
7     'Inform user that the functions have been disabled
8     MsgBox "Sorry! Cutting, copying and pasting have been disabled in this workbook!"
9 End Sub
10
11 Sub EnableMenuItem(ctlId As Integer, Enabled As Boolean)
12     'Activate/Deactivate specific menu item
13     Dim cBar As CommandBar
14     Dim cBarCtrl As CommandBarControl
15     For Each cBar In Application.CommandBars
16         If cBar.Name <> "Clipboard" Then
17             Set cBarCtrl = cBar.FindControl(ID:=ctlId, recursive:=True)
18             If Not cBarCtrl Is Nothing Then cBarCtrl.Enabled = Enabled
19         End If
20     Next
21 End Sub
22
23 Public Sub rxIRibbonUI_onLoad(ByRef ribbon As IRibbonUI)
24     Set grxIRibbonUI_CutCopyPaste = ribbon
25 End Sub
26
27 Public Sub rxshared_getEnabled(ByRef Control As IRibbonControl, ByRef returnedVal)
28     Select Case Control.ID
29     Case "Cut"
30         returnedVal = True 'Set To False to Disable
31     Case "Copy"
32         returnedVal = True 'Set To False to Disable
33     Case "Paste"
34         returnedVal = True 'Set To False to Disable
35     End Select
36 End Sub
37
38 Sub tests()
39     ToggleCutCopyAndPaste (True)
40     Call EnableMenuItem(21, True)
41 End Sub
42
43
44 Sub ToggleCutCopyAndPaste(Allow As Boolean)
45     'Activate/deactivate cut, copy, paste and pastespecial menu items
46     Call EnableMenuItem(21, Allow) ' cut
47     Call EnableMenuItem(19, Allow) ' copy
48     Call EnableMenuItem(22, Allow) ' paste
49     Call EnableMenuItem(755, Allow) ' pastespecial
50
51     'Activate/deactivate drag and drop ability
52     Application.CellDragAndDrop = Allow
53
54     'Activate/deactivate cut, copy, paste and pastespecial shortcut keys
55     With Application
56         Select Case Allow
57         Case Is = False
58             Application.CutCopyMode = False 'Clear Clipboard
59             .OnKey "^c" , "CutCopyPasteDisabled"
60             .OnKey "^v" , "CutCopyPasteDisabled"
```

1 2 3

```
1 2 3
61 .OnKey "^x" , "CutCopyPasteDisabled"
62 .OnKey "+{DEL}" , "CutCopyPasteDisabled"
63 .OnKey "^{INSERT}" , "CutCopyPasteDisabled"
64 } Case Is = True
65 .OnKey "^c"
66 .OnKey "^v"
67 .OnKey "^x"
68 .OnKey "+{DEL}"
69 .OnKey "^{INSERT}"
70 End Select
71 End With
72 End Sub
```

```

1 Option Explicit
2 Dim InvRow As Long, InvItemRow As Long, LastRow As Long, LastItemRow As Long, FieldRow
  As Long, ItemDBRow As Long
3 Dim ResultRow As Long, LastResultRow As Long, ItemRow As Long, TermRow As Long,
  StatusRow As Long, InvNumb As Long
4 Dim HeadShp As Shape
5
6 Sub Customer_AddUpdate()
7     Dim CustRow As Long, CustCol As Long
8     Dim CustomerFld As Control
9     Unload AddCustForm
10    'If Existing Customer is selected, Load Customer Details
11    If Invoice.Range("B3" ).Value <> Empty Then 'Existing Customer
12        CustRow = Invoice.Range("B3" ).Value
13        For CustCol = 2 To 8 'Loop Through Customer Columns
14            Set CustomerFld = AddCustForm.Controls("Field" & CustCol - 1)
15            CustomerFld.Value = Customers.Cells(CustRow, CustCol).Value 'Map Data from
              customers into userform
16        Next CustCol
17    End If
18    AddCustForm.Show
19 End Sub
20 Sub Invoice_Delete()
21     If MsgBox("Are you sure you want to delete this Invoice?" , vbYesNo, "Delete
      Invoice" ) = vbNo Then Exit Sub
22     With Invoice
23         AppEvents_Stop
24         If .Range("B5" ).Value = Empty Then GoTo NotSaved
25         InvRow = .Range("B5" ).Value 'Order Row
26         InvoiceList.Range(InvRow & ":" & InvRow).EntireRow.Delete 'Delete Invoice row
27         'Remove Invoice Items
28         With InvoiceItems
29             LastRow = .Range("A99999" ).End(xlUp).Row
30             If LastRow < 3 Then GoTo NotSaved
31             .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
              "M2:M3" ), CopyToRange:=.Range("P2:W2" ), Unique:=True
32             LastResultRow = .Range("P99999" ).End(xlUp).Row
33             If LastResultRow < 3 Then GoTo NotSaved
34             If LastResultRow < 4 Then GoTo SingleRow
35             'Sort based on descending rows
36             With .Sort
37                 .SortFields.Clear
38                 .SortFields.Add Key:=InvoiceItems.Range("W3" ), SortOn:=xlSortOnValues,
              Order:=xlDescending, DataOption:=xlSortNormal
39                 .SetRange InvoiceItems.Range("P3:W" & LastResultRow)
40             .Apply
41             End With
42             SingleRow:
43             For ResultRow = 3 To LastResultRow
44                 ItemDBRow = .Range("W" & ResultRow).Value 'Invoice Item DB Row
45                 .Range(ItemDBRow & ":" & ItemDBRow).EntireRow.Delete 'Delete Invoice Item
              Row
46             Next ResultRow
47         End With
48         NotSaved:
49         Invoice_New 'Clear Out All Invoice Fields
50         AppEvents_Start
51     End With
52 End Sub
53

```

```

54 Sub Invoice_Load()
55     With Invoice
56         If .Range("B5" ).Value = Empty Then
57             MsgBox "Please enter a correct invoice #"
58             Exit Sub
59         End If
60         AppEvents_Stop
61         InvRow = .Range("B5" ).Value 'Invoice Row
62         .Range("B1" ).Value = True 'Set Inv. Load to true
63         .Range("F2,E4:F9,J10:N44,P10:Q44" ).ClearContents
64         .Range("B11" ).Value = 1 'Set Default To Load First Page
65         For FieldRow = 4 To 9
66             .Range("E" & FieldRow).Value = InvoiceList.Cells(InvRow, FieldRow - 2).Value
67             'Fill Invoice Details
68         Next FieldRow
69         .Range("B12" ).Value = InvoiceList.Range("I" & InvRow).Value 'Set Total Pages
70         Invoice_PageLoad 'Load Page Items
71         .Range("B1" ).Value = False 'Set inv. Load to false
72         AppEvents_Start
73     End With
74 End Sub

75 Sub Invoice_New()
76     With Invoice
77         .Range("B1" ).Value = True 'Set Inv. Load to true
78         .Range("B4,F2,E4:F9,J10:N44,P10:Q44" ).ClearContents
79         .Range("B4" ).Value = .Range("B6" ).Value 'Set Next invoice #
80         .Range("E4" ).Value = Date 'Set current Date
81         .Range("B11,B12" ).Value = 1 'Set Selected Page and Total Pages to 1
82         .Range("B1" ).Value = False 'Set inv. Load to false
83         On Error Resume Next
84         TermRow = Admin.Range("J6:J11" ).Find(Chr(252), , xlValues, xlWhole).Row
85         'Default Payment Term Row
86         On Error GoTo 0
87         If TermRow <> 0 Then .Range("E6" ).Value = Admin.Range("H" & TermRow).Value
88         'Set Default Term
89         On Error Resume Next
90         StatusRow = Admin.Range("F6:F12" ).Find(Chr(252), , xlValues, xlWhole).Row
91         'Default Status Row
92         On Error GoTo 0
93         If StatusRow <> 0 Then .Range("E7" ).Value = Admin.Range("E" & StatusRow).Value
94         'Set Default Status
95         .Range("E5" ).Select
96     End With
97 End Sub

98 Sub Invoice_NextInvoice()
99     With Invoice
100         Dim MaxInvNumb As Long
101         On Error Resume Next
102         MaxInvNumb = Application.WorksheetFunction.Max(InvoiceList.Range("Invoice_ID" ))
103         On Error GoTo 0
104         InvNumb = .Range("B4" ).Value
105         If MaxInvNumb = 0 Then
106             MsgBox "Please create and save an Invoice first"
107             Exit Sub
108         End If
109         If InvNumb = MaxInvNumb Then
110             MsgBox "You are at the last Invoice"
111             Exit Sub

```

1 2 3

```

1 2 3
109 End If
110
111
112 If InvNum = 0 Or .Range("B5").Value = Empty Then 'On New Invoice
113     InvRow = 3 'On Empty Invoice Go to First one created
114 Else 'On Existing Inv. find Next one
115     InvRow = InvoiceList.Range("Invoice_ID").Find(InvNum, , xlValues, xlWhole).
116     Row + 1
117 End If
118 .Range("B4").Value = InvoiceList.Range("A" & InvRow).Value
119 Invoice_Load
120 End With
121 End Sub
122
123
124 Sub Invoice_PrevInvoice()
125     With Invoice
126         Dim MinInvNum As Long
127         On Error Resume Next
128         MinInvNum = Application.WorksheetFunction.Min(InvoiceList.Range("Invoice_ID" ))
129         On Error GoTo 0
130         If MinInvNum = 0 Then
131             MsgBox "Please create and save an Invoice first"
132             Exit Sub
133         End If
134         InvNum = .Range("B4").Value
135         If InvNum = 0 Or .Range("B5").Value = Empty Then 'On New Invoice
136             InvRow = InvoiceList.Range("A99999").End(xlUp).Row 'On Empty Invoice Go to
137             last one created
138         Else 'On Existing Inv. find Previous one
139             InvRow = InvoiceList.Range("Invoice_ID").Find(InvNum, , xlValues, xlWhole).
140             Row - 1
141         End If
142         If .Range("B6").Value = 1 Or MinInvNum = 0 Or MinInvNum = .Range("B4").Value
143             Then
144                 MsgBox "You are at the first invoice"
145                 Exit Sub
146             End If
147             .Range("B4").Value = InvoiceList.Range("A" & InvRow).Value 'Place Inv. ID
148             inside cell
149             Invoice_Load
150         End With
151     End Sub
152
153 Sub Invoice_SavedMsg()
154     With Invoice.Shapes("InvSavedMsg" )
155         Dim i As Long, Delay As Double, StartTime As Double
156         .Visible = msoCTrue
157         For i = 1 To 100
158             .Fill.Transparency = i / 100
159             Delay = 0.009
160             StartTime = Timer
161             Do
162                 DoEvents
163                 Loop While Timer - StartTime < Delay
164             Next i
165         .Visible = msoFalse
166     End With
167 End Sub
168
169 Sub Invoice_SaveUpdate()
170

```

```
1
164 With Invoice
165     If .Range("B3" ).Value = Empty Then
166         MsgBox "Please make sure to add a customer before saving invoice"
167         Exit Sub
168     End If
169     AppEvents_Stop
170     'Deterermine New Invoice/Existing Invoice
171     If .Range("B5" ).Value = Empty Then 'new Invoice
172         InvRow = InvoiceList.Range("A99999" ).End(xlUp).Row + 1 'First Avail Row
173         .Range("B4" ).Value = .Range("B6" ).Value 'Next Inv. #
174         InvoiceList.Range("A" & InvRow).Value = .Range("B4" ).Value 'Next Inv. #
175     Else 'Existing Invoice
176         InvRow = .Range("B5" ).Value 'Invoice Row
177     End If
178     For FieldRow = 4 To 9
179         InvoiceList.Cells(InvRow, FieldRow - 2).Value = .Range("E" & FieldRow).Value
180         'Add Invoice Details
181     Next FieldRow
182     InvoiceList.Range("H" & InvRow).Value = .Range("AD21" ).Value 'Invoice Total
183     InvoiceList.Range("I" & InvRow).Value = .Range("B12" ).Value 'Total Pages
184 End With
185 AppEvents_Start
186 If Application.Caller = "SaveBtn" Then Invoice_SavedMsg 'Run Invoice Fade Out
    Message
End Sub
```

```

1 Option Explicit
2 Dim InvRow As Long, InvItemRow As Long, LastRow As Long, LastItemRow As Long, FieldRow
  As Long, ItemDBRow As Long
3 Dim ResultRow As Long, LastResultRow As Long, ItemRow As Long, TermRow As Long,
  StatusRow As Long, InvNumb As Long
4 Dim LastPage As Long, PageNumb As Long, PageUpd As Long
5 Dim HeadShp As Shape
6
7
8 Sub Invoice_DeletePage()
9     If MsgBox("Are you sure you want to delete this page and all of it's items?" ,
  vbYesNo, "Delete Page" ) = vbNo Then Exit Sub
10    With Invoice
11        If .Range("B11" ).Value = 1 And .Range("B12" ).Value = 1 Then
12            MsgBox "This invoice contains just a single page, and cannot be deleted"
13            Exit Sub
14        End If
15        LastPage = .Range("B12" ).Value 'Last Page Number
16        PageNumb = .Range("B11" ).Value 'Current Page Number
17    End With
18    With InvoiceItems
19        LastRow = .Range("A99999" ).End(xlUp).Row
20        If LastRow < 4 Then GoTo NoItems
21        .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
  "M2:N3" ), CopyToRange:=.Range("P2:W2" ), Unique:=True
22        LastResultRow = .Range("W99999" ).End(xlUp).Row 'Item DB Row
23        If LastResultRow < 3 Then GoTo NoItems
24        If LastResultRow < 4 Then GoTo SkipSort
25        With .Sort 'Sort Descending based on DB Row
26            .SortFields.Clear
27            .SortFields.Add Key:=InvoiceItems.Range("W3" ), SortOn:=xlSortOnValues, Order:
  =xlDescending, DataOption:=xlSortNormal 'Sort
28            .SetRange InvoiceItems.Range("P3:W" & LastResultRow) 'Set Range
29            .Apply 'Apply Sort
30        End With
31        SkipSort:
32        For ResultRow = 3 To LastResultRow
33            ItemDBRow = .Range("W" & ResultRow).Value 'Get Invoice Item DB Row
34            If ItemDBRow <> 0 Then .Range(ItemDBRow & ":" & ItemDBRow).EntireRow.Delete
  'Delete Row
35        Next ResultRow
36        NoItems:
37
38        If PageNumb = LastPage Then GoTo SkipPageUpdate
39        For PageUpd = PageNumb + 1 To LastPage
40            Invoice.Range("B11" ).Value = PageUpd 'Set Page For Filter
41            If LastRow < 4 Then GoTo NoItems
42            .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
  "M2:N3" ), CopyToRange:=.Range("P2:W2" ), Unique:=True
43            LastResultRow = .Range("P99999" ).End(xlUp).Row
44            If LastResultRow < 3 Then GoTo NoPageItems
45            For ResultRow = 3 To LastResultRow
46                ItemDBRow = .Range("W" & ResultRow).Value 'Get Invoice Row
47                .Range("B" & ItemDBRow).Value = .Range("B" & ItemDBRow).Value - 1
  'Reduce Page #
48            Next ResultRow
49            NoPageItems:
50        Next PageUpd
51        SkipPageUpdate:
52        Invoice.Range("B12" ).Value = LastPage - 1 'Reduce total Pages

```

1 2



```

1 2
53   If PageNumb = 1 Then Invoice_NextPage Else Invoice_PrevPage 'Load Previous or
    Next page
54   Invoice_SaveUpdate
55   End With
56 End Sub
57
58 Sub Invoice_NewPage()
59   Invoice_SaveUpdate 'Save New Last page limit
60   Invoice.Range("B11").Value = Invoice.Range("B12").Value + 1 'Set Selected page to
    new page
61   Invoice.Range("B12").Value = Invoice.Range("B12").Value + 1 ' Increment page to +
    1
62   Invoice_PageLoad
63   Invoice_SaveUpdate 'Save New Last page limit
64 End Sub
65
66 Sub Invoice_NextPage()
67   If Invoice.Range("B11").Value = Invoice.Range("B12").Value Then
68     MsgBox "You are on the last page"
69     Exit Sub
70   End If
71   Invoice.Range("B11").Value = Invoice.Range("B11").Value + 1 'Increment Page # Up
72   Invoice_PageLoad
73 End Sub
74
75 Sub Invoice_PageLoad()
76   If Invoice.Range("B5").Value = Empty Then Exit Sub 'Exit on no invoice
77   Invoice.Range("B1").Value = True 'Set Inv. Load to true
78   Invoice.Range("B7,J10:N44,P10:Q44").ClearContents 'Clear Existing Page Items
79   With InvoiceItems
80     LastRow = .Range("A99999").End(xlUp).Row
81     If LastRow < 4 Then GoTo NoItems
82     .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
        "M2:N3"), CopyToRange:=.Range("P2:W2"), Unique:=True
83     LastResultRow = .Range("P99999").End(xlUp).Row
84     If LastResultRow < 3 Then GoTo NoItems
85     For ResultRow = 3 To LastResultRow
86       InvItemRow = .Range("P" & ResultRow).Value 'Get Invoice Row
87       Invoice.Range("J" & InvItemRow & ":N" & InvItemRow).Value = .Range("Q" &
        ResultRow & ":U" & ResultRow).Value 'Item Details
88       Invoice.Range("P" & InvItemRow & ":Q" & InvItemRow).Value = .Range("V" &
        ResultRow & ":W" & ResultRow).Value 'Tax & DB Row
89     Next ResultRow
90     NoItems:
91   End With
92   Invoice.Range("B1").Value = False 'Set inv. Load to false
93 End Sub
94
95 Sub Invoice_PrevPage()
96   If Invoice.Range("B11").Value = 1 Then
97     MsgBox "You are on the first page"
98     Exit Sub
99   End If
100  Invoice.Range("B11").Value = Invoice.Range("B11").Value - 1 'Increment Page #
    Down
101  Invoice_PageLoad
102 End Sub
103
104 Sub Page_DeleteLineItem()
105   Dim SelRow As Long, ResultItemDBRow As Long
1

```

```
1
106 With Invoice
107     If .Range("B7" ).Value = Empty Then Exit Sub
108     SelRow = .Range("B7" ).Value
109     ItemDBRow = .Range("Q" & SelRow).Value 'Invoice Item DB Row
110     LastResultRow = InvoiceItems.Range("P9999" ).End(xlUp).Row 'Last Invoice item
        Row
111     If ItemDBRow = 0 Or LastResultRow < 3 Then Exit Sub
112     For ResultRow = 3 To LastResultRow
113         InvRow = InvoiceItems.Range("P" & ResultRow).Value 'Invoice Row
114         If InvRow > SelRow Then 'Update Invoice Row for Items after selected row
115             ResultItemDBRow = InvoiceItems.Range("W" & ResultRow).Value
116             InvoiceItems.Range("J" & ResultItemDBRow).Value = InvoiceItems.Range("J"
                & ResultItemDBRow).Value - 1 'Reduce Invoice Row
117         End If
118     Next ResultRow
119     InvoiceItems.Range(ItemDBRow & ":" & ItemDBRow).EntireRow.Delete
120     Invoice_PageLoad
121 End With
122 End Sub
```

```

1 Dim SelCell As Range
2 Dim DayName As String
3
4
5
6 .....''Color Calendar Background''.....
7 Sub CalCol()
8     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
9         With Selection.ShapeRange.Fill
10            .ForeColor.RGB = ActiveSheet.Shapes(Application.Caller).Fill.ForeColor.RGB
11        End With
12        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
13    End With
14 End Sub
15 Sub CalendarHide()
16     Dim DayNum As Long
17     'Hide Calendar, Reset Day Colors
18     On Error GoTo NoCal
19     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
20     Sheets("CalPopUp" ).Range("A7" ).Value = ""
21     If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
22         For DayNum = 1 To 42
23             DayName = DayNum & "Day"
24             With ActiveSheet.Shapes(DayName)
25                 .Fill.ForeColor.RGB = RGB(255, 255, 255)
26                 .TextFrame2.TextRange.Font.Bold = msoFalse
27             End With
28         Next DayNum
29     End If
30     Exit Sub
31     NoCal: 'If calendar has been removed by accident, paste in backup calendar from
           CalPopUp Sheet
32 End Sub
33
34 Sub CalendarShow()
35     With ActiveSheet
36         Set SelCell = ActiveCell
37         'Check if active cell is a valid date
38         If IsDate(SelCell.Value) = True Then
39             Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
40         Else: 'If No Date or incorrect Date user current date
41             Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
42         End If
43         'Clear all shapes to white (if calendar is visible)
44         If ActiveSheet.Shapes("Calendar" ).Visible = True Then
45             For DayNum = 1 To 42
46                 DayName = DayNum & "Day"
47                 With ActiveSheet.Shapes(DayName)
48                     .Fill.ForeColor.RGB = RGB(255, 255, 255)
49                     .TextFrame2.TextRange.Font.Bold = msoFalse
50                 End With
51             Next DayNum
52         End If
53
54         Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).
           Value) 'Set Month
55         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
           Value) 'Set Year
56         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
57         ' UnGroupCal

```

1 2

```

1 2
58 If InStr(.Shapes("1Day").DrawingObject.Formula, "J" ) <> 0 Then 'Run Workbook
Link Remover and Cell Link Replacement
59     MacroLinkRemover
60     CalFormulaReplacement
61 End If
62 'GroupCal
63 On Error GoTo NoCal
64 If DayName = "Day" Then DayName = Day(Date) & "Day" 'Set Default Day
65 With ActiveSheet.Shapes(DayName)
66     .Fill.ForeColor.RGB = RGB(252, 213, 180)
67     .TextFrame2.TextRange.Font.Bold = msoTrue
68 End With
69 On Error GoTo NoCal
70 .Shapes("Calendar").Visible = msoCTrue
71 .Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3", "CalCol4",
"CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible = False '
72 .Shapes("Calendar").Left = SelCell.Left
73 .Shapes("Calendar").Placement = xlMove
74 .Shapes("Calendar").Top = SelCell.Offset(1, 0).Top
75 If Sheets("CalPopUp").Range("A6").Value > 0 Then
76     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day"
, "42Day")).Visible = True
77 Else:
78     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day"
, "42Day")).Visible = False
79 End If
80 Sheets("CalPopUp").Range("A7").Value = SelCell.Address
81 ActiveCell.Select
82 End With
83 Exit Sub
84 NoCal:
85 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
calendar over from another sheet and paste into this sheet"
86 End Sub
87 Sub CalFormulaReplacement()
88 With ActiveSheet
89     Dim DayNum, ColNum, RowNum As Long
90     Dim Shp As Shape
91     ColNum = 2
92     RowNum = 1
93     For DayNum = 1 To 42
94         .Shapes(DayNum & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(RowNum
, ColNum).Address
95         ColNum = ColNum + 1
96         If ColNum = 9 Then
97             ColNum = 2
98             RowNum = RowNum + 1
99         End If
100     Next DayNum
101     .Shapes("Month").DrawingObject.Formula = "=CalPopUp!$A$4"
102     .Shapes("Year").DrawingObject.Formula = "=CalPopUp!$A$2"
103 End With
104 End Sub
105
106 Sub CheckForSheet()
107 'Checks for existance of Calendar Pop-up Worksheet
108 Dim ws As Worksheet
109 On Error GoTo CreateWS
110 Set ws = ActiveWorkbook.Sheets("CalPopUp")
111 Exit Sub

```

```

1
112 CreateWS:
113 CreateCalSht
114 End Sub
115
116
117 'Create Calendar Sheet on First Run of Calendar
118 Sub CreateCalSht()
119 Dim ColCnt, RowCnt, DayCnt, CalCol As Long
120 Dim ws, ActSht As Worksheet
121 Set ActSht = ActiveSheet
122 'On Error GoTo NoCal
123 ActiveSheet.Shapes("Calendar").Copy
124 Set ws = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.
Count))
125 ws.Name = "CalPopUp"
126 ActSht.Activate
127
128 'Reassign Shape Links & Macros
129 With ActiveSheet
130 UnGroupCal
131 Unprotect
132 .Shapes("PrevYr").OnAction = "" & ActiveWorkbook.Name & "!PrevYear"
133 .Shapes("NextYr").OnAction = "" & ActiveWorkbook.Name & "!NextYear"
134 .Shapes("NextRec").OnAction = "" & ActiveWorkbook.Name & "!NextMonth"
135 .Shapes("NextTri").OnAction = "" & ActiveWorkbook.Name & "!NextMonth"
136 .Shapes("PrevRec").OnAction = "" & ActiveWorkbook.Name & "!PrevMonth"
137 .Shapes("PrevTri").OnAction = "" & ActiveWorkbook.Name & "!PrevMonth"
138 .Shapes("SetBtn").OnAction = "" & ActiveWorkbook.Name & "!ShowSettings"
139 .Shapes("Month").DrawingObject.Formula = "=CalPopUp!A4"
140 .Shapes("Year").DrawingObject.Formula = "=CalPopUp!A2"
141 DayCnt = 1
142 For RowCnt = 1 To 6
143 For ColCnt = 2 To 8
144 .Shapes(DayCnt & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(
RowCnt, ColCnt).Address 'Assigned Linked Cell
145 .Shapes(DayCnt & "Day").OnAction = "" & ActiveWorkbook.Name &
"!SelectDay" 'Assign Macro
146 DayCnt = DayCnt + 1
147 Next ColCnt
148 Next RowCnt
149
150 'Assign Color Macros
151 For CalCol = 1 To 9
152 .Shapes("CalCol" & CalCol).OnAction = "" & ActiveWorkbook.Name &
"!CalCol" 'Assign Color Macro
153 Next CalCol
154 End With
155
156 With Sheets("CalPopUp")
157 .Paste
158 .Visible = xlSheetHidden
159
160 'Add in Formulas and Details
161 .Range("A1").Value = Date 'Set Current Date
162 .Range("A2").Value = Year(Date) 'Set Current Year
163 .Range("A3").Value = Month(Date) 'Set Current Month #
164 .Range("A4").Value = "=INDEX(CalMonths,A3,)"
165 .Range("A5").Value = "=A4&" & Chr(34) & " " & Chr(34) & "&CalYear"
166 .Range("A6").Value = "=SUM(B6:H6)"
167 .Range("A8").Value = "January"

```

1 2

```

1 2
168 .Range("A8" ).AutoFill Destination:=.Range("A8:A19" ), Type:=xlFillDefault
169 .Range("A20" ).Value =
    "=IFERROR(INDIRECT(ADDRESS(SUMPRODUCT((B1:H6=A1)*ROW(B1:H6))+6,SUMPRODUCT((B1:H6=A
1)*COLUMN(B1:H6)),1,1))," & Chr(34) & Chr(34) & ")"
170 'Set Defined Names
171 ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
172 ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
173
174 'Add in Calendar Formulas
175
176 .Range("B1" ).Value =
    "=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,Cal
177 Months,0),1)," & Chr(34) & Chr(34) & ")"
    .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
    ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A
178 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
    .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
    ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A
179 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
    .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
    ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A
180 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
    .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
    ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A
181 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
    .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
    ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A
182 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
    .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
    ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A
183 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
    .Range("B2" ).Value = "=H1+1"
184 .Range("C2" ).Value = "=B2+1"
185 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
186 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
187 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
    & Chr(34) & Chr(34) & ",H4+1)"
188 .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",B5+1)," & Chr(34) & Chr(34) & ")"
189 .Range("B6" ).Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) &
    ",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) &
    ")")
190 .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) &
    ",I5+1)," & Chr(34) & Chr(34) & ")"
191 .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
192
193 'Set format to Single Day
194 .Range("B1:H6" ).NumberFormat = "d"
195
196 'Add in relative Day #'s
197 .Range("B7" ).Value = "1"
198 .Range("C7" ).Value = "2"
199 .Range("B8" ).Value = "8"
200 .Range("C8" ).Value = "9"
201 .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
202 .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
203 GroupCal
204 End With
205 Exit Sub
206 NoCal:

```

```

1
207 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
calendar over from another sheet and paste into this sheet"
208 End Sub
209 Sub GroupCal()
210 ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
211 Selection.ShapeRange.Name = "NextMonth"
212 ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
213 Selection.ShapeRange.Name = "PrevMonth"
214 ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
215 , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" ,
"1Day" , _
216 "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" ,
"6Day" , _
217 "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
"18Day" _
218 , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day"
, _
219 "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
"Sa" , _
220 "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
"CalCol3" , _
221 "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" ,
"PrevMonth" , _
222 "NextMonth" , "NextYr" , "PrevYr" ).Visible = msoCTrue
223 ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
224 , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day"
, _
225 "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
226 "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
"18Day" _
227 , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
228 "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" ,
"Sa" , _
229 "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
"CalCol3" , _
230 "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth"
, _
231 "NextMonth" , "NextYr" , "PrevYr" )).Select
232 Selection.ShapeRange.Group.Select
233 Selection.ShapeRange.Name = "Calendar"
234 Selection.Name = "Calendar"
235 Selection.Placement = xlMove
236 ActiveSheet.Shapes("Calendar" ).Placement = 2
237 End Sub
238
239
240
241
242 Sub MacroLinkRemover()
243 'PURPOSE: Remove an external workbook reference from all shapes triggering macros
244 'Source: www.ExcelForFreelancers.com
245 Dim Shp As Shape
246 Dim MacroLink, NewLink As String
247 Dim SplitLink As Variant
248
249 For Each Shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
250
1 2

```

```

1 2
251 'Grab current macro link (if available)
252 On Error GoTo NextShp
253 MacroLink = Shp.OnAction
254
255 'Determine if shape was linking to a macro
256 If MacroLink <> "" And InStr(MacroLink, "!") <> 0 Then
257     'Split Macro Link at the exclamation mark (store in Array)
258     SplitLink = Split(MacroLink, "!")
259
260     'Pull text occurring after exclamation mark
261     NewLink = SplitLink(1)
262
263     'Remove any straggling apostrophes from workbook name
264     If Right(NewLink, 1) = "'" Then
265         NewLink = Left(NewLink, Len(NewLink) - 1)
266     End If
267
268     'Apply New Link
269     Shp.OnAction = NewLink
270 End If
271 NextShp:
272 Next Shp
273 End Sub
274 Sub NextMonth()
275     'Next Month button
276     If Sheets("CalPopUp").Range("A20").Value <> Empty Then
277         DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
278         With ActiveSheet.Shapes(DayName)
279             .Fill.ForeColor.RGB = RGB(255, 255, 255)
280             .TextFrame2.TextRange.Font.Bold = msoFalse
281         End With
282     End If
283     With Sheets("CalPopUp")
284         If .Range("A3").Value = 12 Then
285             .Range("A3").Value = 1
286             .Range("A2").Value = .Range("A2").Value + 1
287         Else:
288             .Range("A3").Value = .Range("A3").Value + 1
289         End If
290         If .Range("A6").Value > 0 Then
291             ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day",
292             "40Day", "41Day", "42Day")).Visible = True
293         Else:
294             ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day",
295             "40Day", "41Day", "42Day")).Visible = False
296         End If
297     End With
298 End Sub
299 Sub NextYear()
300     ThisWorkbook.Sheets("CalPopUp").Range("A2").Value = ThisWorkbook.Sheets(
301     "CalPopUp").Range("A2").Value + 1
302 End Sub
303 Sub PrevMonth()
304     'Previous Month Button
305     If Sheets("CalPopUp").Range("A20").Value <> Empty Then
306         DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
307         With ActiveSheet.Shapes(DayName)
308             .Fill.ForeColor.RGB = RGB(255, 255, 255)
309             .TextFrame2.TextRange.Font.Bold = msoFalse
310         End With
311     End If
312 End Sub
1 2 3

```



```

1 2 3
308     End With
309     End If
310     With Sheets("CalPopUp" )
311         If .Range("A3" ).Value = 1 Then
312             .Range("A3" ).Value = 12
313             .Range("A2" ).Value = .Range("A2" ).Value - 1
314         Else:
315             .Range("A3" ).Value = .Range("A3" ).Value - 1
316         End If
317         If .Range("A6" ).Value > 0 Then
318             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
319             "40Day" , "41Day" , "42Day" )).Visible = True
320         Else:
321             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
322             "40Day" , "41Day" , "42Day" )).Visible = False
323         End If
324     End With
325 End Sub
326
327 Sub PrevYear()
328     ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
329     "CalPopUp" ).Range("A2" ).Value - 1
330 End Sub
331
332 Sub ReplaceCalendar() 'Shape Deleted
333     ThisWorkbook.Sheets("CalPopUp" ).Shapes("Calendar" ).Copy 'Copy From Developers
334     sheet
335     ActiveCell.Select 'Select the active Cell
336     ActiveSheet.Paste 'Paste in Calendar
337 End Sub
338
339 .....
340
341 ''''Select Day Of The Month
342 .....
343
344 Sub SelectDay()
345     Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
346     DayNumb = Replace(Application.Caller, "Day" , "" )
347     RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
348     ColNumb = DayNumb Mod 7 + 1
349     If ColNumb = 1 Then ColNumb = 8
350     'On Error Resume Next
351     If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
352     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
353     ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
354     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
355     ActiveCell.Offset(0, 1).Select
356 End Sub
357
358 Sub ShowSettings()
359     'Show or Hide Calendar Settings Panel
360     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
361         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
362         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
363         = False
364     Else:
365         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
366         "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
367         = True
368     End If
369 End Sub
370
371 Sub UnGroupCal()
372

```

```
1
359 On Error Resume Next
360 ActiveSheet.Shapes("Calendar").Ungroup
361 ActiveSheet.Shapes("NextMonth").Ungroup
362 ActiveSheet.Shapes("PrevMonth").Ungroup
363 On Error GoTo 0
364 End Sub
```

```
1 Option Explicit
2 Dim LastPage As Long, PageNumb As Long
3 Sub Invoice_Print()
4     With Invoice
5         Invoice_SaveUpdate 'Save Current Invoice
6         LastPage = .Range("B12").Value 'Set Last Page
7         For PageNumb = 1 To LastPage
8             .Range("B11").Value = PageNumb
9             Invoice_PageLoad 'Load Page
10            .PrintOut , , , False, True, , , , False
11        Next PageNumb
12    End With
13 End Sub
14
15 Sub Invoice_SaveAsPDF()
16     Dim Pages As String, FilePath As String, SheetsArr() As String
17     Application.ScreenUpdating = False
18     With Invoice
19         LastPage = .Range("B12").Value 'Set Last Page
20         FilePath = ThisWorkbook.Path & "Invoice.pdf"
21         For PageNumb = 1 To LastPage
22             .Range("B11").Value = PageNumb
23             Invoice_PageLoad 'Load Page
24             .Select
25             .Copy After:=Sheets(Sheets.Count)
26             ActiveSheet.Name = "Page" & PageNumb
27             If PageNumb <> LastPage Then
28                 Pages = Pages & "Page" & PageNumb & "," 'Build Sheet name string
29             Else
30                 Pages = Pages & "Page" & PageNumb 'no comma on last page
31             End If
32         Next PageNumb
33         SheetsArr = Split(Pages, ",") 'Create Sheet array
34         Sheets(SheetsArr).Select
35         ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, Filename:=FilePath, _
36             Quality:=xlQualityStandard, IncludeDocProperties:=True,
37             IgnorePrintAreas _
38                 :=False, OpenAfterPublish:=True
39         'Delete Temporary Sheets
40         Sheets(SheetsArr).Select
41         Application.DisplayAlerts = False
42         Sheets(SheetsArr).Delete
43         Application.DisplayAlerts = True
44         .Activate
45     End With
46     Application.ScreenUpdating = True
47 End Sub
```

```

1  Option Explicit
2
3  Sub DisableDragAndDrop()
4      Application.CellDragAndDrop = False
5  End Sub
6
7  Sub HideNamedRange()
8      ThisWorkbook.Names("UserPW" ).Visible = False
9  End Sub
10
11  'Hide Sheet (standard)
12  Sub HideSheet_Hidden()
13      CalPopUp.Visible = xlSheetHidden
14  End Sub
15
16
17  Sub HideSheet_VeryHidden()
18      CalPopUp.Visible = xlSheetVeryHidden
19  End Sub
20
21  Sub ListAllSheets()
22      Dim WorkSht As Worksheet
23      Dim SheetRow As Long
24      With Protection
25          .Range("N5:N999" ).ClearContents 'Clear Existing Sheet details
26          SheetRow = 5 'Set Initial Row
27          For Each WorkSht In ThisWorkbook.Worksheets 'Loop through all sheets within the
                workbook
28              .Range("N" & SheetRow).Value = WorkSht.Name 'WorkSheet name
29              SheetRow = SheetRow + 1 'Increment Sheet Row
30          Next WorkSht
31      End With
32  End Sub
33
34
35  Sub LoopThroughNamedRanges()
36      Dim NmRange As Name
37      For Each NmRange In ThisWorkbook.Names
38          If NmRange.Visible = False Then NmRange.Visible = True
39      Next NmRange
40  End Sub
41
42  Sub Protection_WorksheetDetails()
43      Dim SheetRow As Long, LastRow As Long, SheetCol As Long
44      Dim SheetNm As String, UserType As String, SecLevel As String
45      UserType = Users.Range("B7" ).Value 'Set User type
46      If UserType = Empty Then Exit Sub
47      Start.Activate 'Start sheet, visible to every user, should be set initially
48      With Protection
49          On Error Resume Next
50          SheetCol = .Range("O4:R4" ).Find(UserType, , xlValues, xlWhole).Column
                'Determine User Column
51          On Error GoTo 0
52          If SheetCol = 0 Then Exit Sub
53          LastRow = .Range("N99999" ).End(xlUp).Row
54          If LastRow < 5 Then Exit Sub
55          For SheetRow = 5 To LastRow 'Start our loop
56              SheetNm = .Range("N" & SheetRow).Value 'Sheet Name
57              SecLevel = .Cells(SheetRow, SheetCol).Value 'Extract security level
58              Select Case SecLevel

```

1 2 3

```

1 2 3
59 Case Is = Chr(208) 'Unprotected & Visible
60 Sheets(SheetNm).Unprotect 'Unlocked sheet
61 Sheets(SheetNm).Visible = xlSheetVisible 'Make sure sheet is visible
62 Case Is = Chr(207) 'Protected & Visible
63 Sheets(SheetNm).Protect Password:="", DrawingObjects:=True, Contents:=True
64 , _
65 AllowFormattingCells:=False, _
66 AllowFormattingColumns:=False, AllowFormattingRows:=False,
67 AllowInsertingColumns:=False, _
68 AllowInsertingRows:=False, AllowInsertingHyperlinks:=True,
69 AllowDeletingColumns:=False, _
70 AllowDeletingRows:=False, AllowSorting:=True, AllowFiltering:=True,
71 AllowUsingPivotTables:=True 'Protect Sheet
72 Sheets(SheetNm).Visible = xlSheetVisible 'Make sure sheet is visible
73 Case Is = Chr(120) 'Protected & Hidden
74 Sheets(SheetNm).Protect Password:="", DrawingObjects:=True, Contents:=True
75 , _
76 AllowFormattingCells:=False, _
77 AllowFormattingColumns:=False, AllowFormattingRows:=False,
78 AllowInsertingColumns:=False, _
79 AllowInsertingRows:=False, AllowInsertingHyperlinks:=True,
80 AllowDeletingColumns:=False, _
81 AllowDeletingRows:=False, AllowSorting:=True, AllowFiltering:=True,
82 AllowUsingPivotTables:=True 'Protect Sheet
83 Sheets(SheetNm).Visible = xlVeryHidden ' Hide sheet with very hidden
84 End Select
85 Next SheetRow
86 End With
87 End Sub
88
89
90 Sub ProtectWorksheet()
91 Invoice.Protect Password:="", DrawingObjects:=True, Contents:=True, Scenarios:=True
92 , UserInterfaceonly:=True, AllowFormattingCells:=False, _
93 AllowFormattingColumns:=False, AllowFormattingRows:=False, AllowInsertingColumns:=
94 False, AllowInsertingRows:=False, _
95 AllowInsertingHyperlinks:=True, AllowDeletingColumns:=False, AllowDeletingRows:=
96 False, AllowSorting:=True, AllowFiltering:=True, AllowUsingPivotTables:=True
97 End Sub
98
99 Sub ShowSheet()
100 CalPopUp.Visible = xlSheetVisible
101 End Sub
102
103 Sub ToggleWorksheetTabs(Show As Boolean)
104 ActiveWindow.DisplayWorkbookTabs = Show
105 End Sub
106
107 Sub ToggleWorksheetTabs()
108 ToggleWorksheetTabs (True)
109 End Sub
110
111 Sub User_MachineCheck()
112 Dim MachineID As String, ScriptToRun As String
113 'Check Machine ID
114 #If Mac Then 'Run Script if Mac
115 ScriptToRun = _

```

```
1
108 "set uuid to do shell script " "system_profiler SPHardwareDataType" & " | awk
    '/Hardware UUID:/ {print $NF}'" ""
109 On Error Resume Next
110 MachineID = MacScript(ScriptToRun)
111 On Error GoTo 0
112 #Else 'On PC
113 MachineID = CreateObject("Scripting.FileSystemObject").GetDrive("C:\").
    SerialNumber 'I am Windows
114 #End If
115 If Users.Range("B6").Value = Empty Then 'First Use, Set Machine ID
116     Users.Range("B6").Value = MachineID
117 Else
118     If Users.Range("B6").Value <> MachineID Then 'Non-Matching
119         If Users.Range("B6").Value = "Developer" Then
120             Exit Sub 'Check for Admin/Developer Registration
121         Else
122             MsgBox "Please register the application for this specific machine"
123         End If
124     Else 'Correct ID
125         Exit Sub
126     End If
127 End If
128 End Sub
129
130
131
132 Sub UserLogin()
133     Users.Range("B1,B2,B4").ClearContents 'Clear Current user plus any usernames or
    passwords
134     LoginForm.Username.Value = ""
135     LoginForm.Password.Value = ""
136     LoginForm.Show
137 End Sub
138
139 Sub UserLogout()
140     Dim WkSht As Worksheet
141     Users.Range("B4").ClearContents 'Clear Active User
142     Sheets("Start").Activate
143     For Each WkSht In ThisWorkbook.Worksheets
144         If WkSht.Name <> "Start" Then WkSht.Visible = xlSheetVeryHidden
145     Next WkSht
146 End Sub
```

\_, 9, 35, 39, 41

## A

Activate, 20, 33, 39, 40, 42  
 ActiveCell, 31, 32, 37  
 ActiveSheet, 31-33, 35-39  
 ActiveWindow, 41  
 ActiveWorkbook, 32-34  
 ActSht, 33  
 Add, 21, 24, 28, 33, 34  
 AddCustForm, 18, 24  
 AddItemForm, 6, 19  
 Address, 9, 10, 32, 33  
 Admin, 7, 9, 21, 25  
 AdvancedFilter, 24, 28, 29  
 After, 33, 39  
 Allow, 22  
 AllowDeletingColumns, 41  
 AllowDeletingRows, 41  
 AllowFiltering, 41  
 AllowFormattingCells, 41  
 AllowFormattingColumns, 41  
 AllowFormattingRows, 41  
 AllowInsertingColumns, 41  
 AllowInsertingHyperlinks, 41  
 AllowInsertingRows, 41  
 AllowMultiSelect, 21  
 AllowSorting, 41  
 AllowUsingPivotTables, 41  
 AppEvents\_Start, 21, 24, 25, 27  
 AppEvents\_Stop, 21, 24, 25, 27  
 Application, 6, 18, 21, 22, 25-27, 31, 37, 39, 40  
 Apply, 24, 28  
 Array, 10, 31, 32, 35-37  
 AutoFill, 34

## B

Bold, 31, 32, 36  
 BtnMov, 21

## C

Cal, 7, 8  
 CalCol, 31, 33  
 Calculation, 21  
 CalendarHide, 8, 31  
 CalendarShow, 8, 31  
 CalFormulaReplacement, 32  
 Caller, 27, 31, 37  
 CalPopUp, 40, 41  
 Cancel, 16, 20  
 CancelBtn\_Click, 18-20  
 cBar, 22  
 cBarCtrl, 22  
 CellDragAndDrop, 22, 40  
 Cells, 7-10, 18, 19, 24, 25, 27, 32, 33, 37, 40  
 CheckForLbl, 9, 10  
 CheckForSheet, 8, 32  
 Chr, 3, 8, 9, 14, 19, 25, 33, 34, 41  
 Clear, 24, 28  
 ClearContents, 3, 6, 8, 9, 20, 25, 29, 40, 42  
 ColCnt, 33  
 ColNum, 32  
 ColNumb, 37  
 Column, 7, 40  
 Columns, 9  
 ColumnWidth, 9  
 CommandBar, 22

CommandBarControl, 22  
 CommandBars, 22  
 Company\_AddLogo, 21  
 Contents, 41  
 Control, 18, 19, 22, 24  
 Controls, 18, 19, 24  
 Copy, 33, 37, 39  
 CopyToRange, 24, 28, 29  
 Count, 33, 39  
 CountLarge, 3, 6, 8, 14  
 CreateCalSht, 33  
 CreateObject, 42  
 CreateWS, 32, 33  
 CriteriaRange, 24, 28, 29  
 ctIId, 22  
 CustCol, 18, 24  
 Customer\_AddUpdate, 24  
 CustomerFld, 18, 24  
 Customers, 18, 24  
 CustRow, 18, 24  
 CutCopyMode, 22  
 CutCopyPasteDisabled, 22

## D

DataOption, 24, 28  
 Day, 32  
 DayCnt, 33  
 DayName, 31, 32, 36  
 DayNum, 31, 32  
 DayNumb, 37  
 Delay, 26  
 Delete, 24, 28, 30, 39  
 DescWidth, 8, 9  
 Destination, 34  
 Dir, 9  
 DisableDragAndDrop, 40  
 DisplayAlerts, 39  
 DisplayWorkbookTabs, 41  
 DoEvents, 26  
 DrawingObject, 9, 10, 32, 33  
 DrawingObjects, 41  
 Duplicate, 9, 10

## E

EditMode, 20, 21  
 Empty, 3, 6-10, 14, 18, 19, 24-27, 29-31, 36, 37, 40, 42  
 Enabled, 22  
 EnableEvents, 6, 21  
 EnableMenuItem, 22  
 EntireColumn, 9  
 EntireRow, 24, 28, 30  
 Explicit, 3-6, 11-22, 24, 28, 39, 40  
 ExportAsFixedFormat, 39

## F

Field1, 6, 18, 19  
 FieldRow, 24, 25, 27, 28  
 FileDialog, 21  
 Filename, 39  
 FilePath, 39  
 Fill, 21, 26, 31, 32, 36  
 Filters, 21  
 Find, 6, 7, 25, 26, 40  
 FindControl, 22  
 Font, 31, 32, 36  
 ForeColor, 21, 31, 32, 36  
 Formula, 9, 10, 32, 33

## G

GetDrive, 42  
 Group, 10, 35  
 GroupCal, 34, 35  
 GrpShp, 7, 10  
 grxlRibbonUI\_CutCopyPaste, 22

## H

HeadShp, 24, 28  
 Hidden, 9  
 Hide, 18-20  
 HideNamedRange, 40  
 HideSheet\_Hidden, 40  
 HideSheet\_VeryHidden, 40

## I

i, 26  
 ID, 22  
 IgnorePrintAreas, 39  
 IncludeDocProperties, 39  
 Insert, 9  
 InStr, 32, 36  
 Intersect, 3, 6-8, 14  
 InvItemRow, 6, 7, 24, 28, 29  
 InvNumb, 24-26, 28  
 Invoice, 18-21, 24-30, 39, 41  
 Invoice\_Delete, 24  
 Invoice\_DeletePage, 28  
 Invoice\_Load, 6, 25, 26  
 Invoice\_New, 24, 25  
 Invoice\_NewPage, 29  
 Invoice\_NextInvoice, 25  
 Invoice\_NextPage, 29  
 Invoice\_PageLoad, 25, 29, 30, 39  
 Invoice\_PrevInvoice, 26  
 Invoice\_PrevPage, 29  
 Invoice\_Print, 39  
 Invoice\_SaveAsPDF, 39  
 Invoice\_SavedMsg, 26, 27  
 Invoice\_SaveUpdate, 26, 29, 39  
 InvoiceItems, 7, 24, 28-30  
 InvoiceList, 24-27  
 InvRow, 24-28, 30  
 InvWidth, 8, 9  
 IRibbonControl, 22  
 IRibbonUI, 22  
 IsDate, 31  
 ItemCol, 6-10, 19  
 ItemDBRow, 6, 24, 28, 30  
 ItemFld, 19  
 ItemLblShp, 7, 8, 10  
 ItemNm, 7, 9, 10  
 ItemRow, 6, 9, 10, 19, 24, 28  
 Items, 6, 19  
 ItemShp, 7-10  
 ItemTypeRow, 6, 7, 9

## K

Key, 24, 28

## L

LastItemRow, 24, 28  
 LastPage, 28, 39  
 LastResultRow, 24, 28-30  
 LastRow, 24, 28, 29, 40  
 Left, 8-10, 21, 32, 36  
 Len, 36  
 ListAllSheets, 40  
 LoginForm, 20, 42

LogoPic, 21  
LoopThroughNamedRanges, 40

**M**

Mac, 41  
MachineID, 41, 42  
MacroLink, 35, 36  
MacroLinkRemover, 32, 35  
MacScript, 42  
Max, 18, 25  
MaxInvNumb, 25  
Min, 26  
MinInvNumb, 26  
Month, 31, 33  
MovInc, 21  
MSForms, 20  
MsgBox, 6, 18-22, 24-29, 32, 35, 42  
msoCTrue, 7-10, 26, 32, 35  
msoFalse, 7, 9, 10, 26, 31, 36, 37  
msoFileDialogFilePicker, 21  
msoTrue, 32

**N**

Name, 9, 10, 20, 22, 33-35, 39, 40, 42  
Names, 34, 40  
NewLink, 35, 36  
NextMonth, 36  
NextShp, 36  
NextYear, 36  
NmRange, 40  
NoCal, 31, 32, 34  
NoItems, 28, 29  
NoPageItems, 28  
NoSelection, 21  
NotSaved, 24  
NumberFormat, 34

**O**

Offset, 10, 32, 37  
OkBtn\_Click, 20  
OnAction, 33, 36  
OnKey, 22, 23  
OpenAfterPublish, 39  
Order, 24, 28

**P**

Page\_DeleteLineItem, 29  
PageNumb, 28, 29, 39  
Pages, 39  
PageUpd, 28  
Password, 20, 41, 42  
Password\_Exit, 20  
Paste, 33, 37  
Path, 39  
PicPath, 21  
Pictures, 9  
Placement, 9, 10, 32, 35  
PrevMonth, 36  
PrevYear, 37  
PrintOut, 39  
Protect, 41  
Protection, 40  
Protection\_WorksheetDetails, 40  
ProtectWorksheet, 41

**Q**

Quality, 39

**R**

Range, 3, 6-10, 14, 18-21, 24-37, 39, 40, 42

recursive, 22  
RefersTo, 34  
Replace, 37  
ReplaceCalendar, 8, 37  
ResultItemDBRow, 29, 30  
ResultRow, 24, 28-30  
ReturnBoolean, 20  
returnedVal, 22  
RGB, 21, 31, 32, 36  
ribbon, 22  
Right, 36  
RoundUp, 37  
Row, 3, 6-9, 14, 18, 19, 24-30, 40  
RowCnt, 33  
RowNum, 32  
RowNumb, 37  
rxIRibbonUI\_onLoad, 22  
rxshared\_getEnabled, 22

**S**

SaveAsUI, 16  
SaveBtn\_Click, 18, 19  
Scenarios, 41  
ScreenUpdating, 21, 39  
ScriptToRun, 41, 42  
SecLevel, 40  
SelCell, 31, 32  
SelectDay, 37  
SelectedItems, 21  
Selection, 31, 35  
SelRow, 6, 7, 19, 29, 30  
SerialNumber, 42  
SetRange, 24, 28  
Sh, 16  
Shape, 7, 24, 28, 32, 35  
ShapeRange, 31, 35  
Shapes, 7-10, 21, 26, 31-33, 35-38  
SheetCol, 40  
SheetNm, 40, 41  
SheetRow, 40, 41  
Sheets, 31-33, 36, 37, 39, 41, 42  
SheetsArr, 39  
Show, 6, 21, 24, 41, 42  
ShowSettings, 37  
ShowSheet, 41  
Shp, 32, 35, 36  
SingleRow, 24  
SkipPageUpdate, 28  
SkipSort, 28  
Sort, 24, 28  
SortFields, 24, 28  
SortOn, 24, 28  
Split, 36, 39  
SplitLink, 35, 36  
Start, 40  
StartTime, 26  
StatusRow, 24, 25, 28

**T**

Target, 3, 6-10, 14  
TaxField, 19  
TermRow, 6, 7, 24, 25, 28  
tests, 22  
Text, 20, 21  
TextFrame2, 21, 31, 32, 36  
TextRange, 21, 31, 32, 36  
ThisWorkbook, 20, 33, 36, 37, 39, 40, 42  
Timer, 26  
Title, 21  
ToggleCutCopyAndPaste, 22

ToggleWorksheetTabs, 41  
ToogleWorksheetTabs, 41  
Top, 8-10, 32  
Transparency, 26

**U**

Ungroup, 10, 38  
UnGroupCal, 33, 37  
Unique, 24, 28, 29  
Unload, 24  
Unprotect, 41  
User\_MachineCheck, 41  
UserInterfaceonly, 41  
UserLogin, 42  
UserLogout, 42  
Username, 20, 42  
Username\_Exit, 20  
Users, 20, 21, 40, 42  
UserType, 40

**V**

Value, 3, 6-10, 14, 18-21, 24-34, 36, 37, 39, 40, 42  
vbDirectory, 9  
vbNo, 6, 24, 28  
vbYesNo, 6, 24, 28  
Visible, 7-10, 20, 26, 31-33, 35-37, 40-42

**W**

WkSht, 20, 42  
Workbook\_Activate, 16  
Workbook\_BeforeClose, 16  
Workbook\_BeforeSave, 16  
Workbook\_Deactivate, 16  
Workbook\_Open, 16  
Workbook\_SheetDeactivate, 16  
Worksheet, 20, 32, 33, 40, 42  
Worksheet\_Change, 6  
Worksheet\_SelectionChange, 3, 7, 14  
WorksheetFunction, 18, 25, 26, 37  
Worksheets, 20, 40, 42  
WorkSht, 40  
ws, 32, 33

**X**

xlCalculationAutomatic, 21  
xlCalculationManual, 21  
xlDescending, 24, 28  
xlFillDefault, 34  
xlFilterCopy, 24, 28, 29  
xlMove, 9, 10, 32, 35  
xlQualityStandard, 39  
xlSheetHidden, 33, 40  
xlSheetVeryHidden, 20, 40, 42  
xlSheetVisible, 20, 41  
xlSortNormal, 24, 28  
xlSortOnValues, 24, 28  
xlTypePDF, 39  
xlUp, 7, 18, 19, 24, 26-30, 40  
xlValues, 6, 7, 25, 26, 40  
xlVeryHidden, 41  
xlWhole, 6, 7, 25, 26, 40

**Y**

Year, 31, 33



# 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*