

# VBA SOURCE CODE BOOK

## INVOICE PAYMENT MANAGER

**INVOICE PAYMENTS MANAGER**

Customer: Dolores Richman      Payment Date: 1/5/2023

Payment Type: Bank Transfer      Pmnt. Amount: \$5,066.22

Payment Notes: Final Payment Due in 30 Days

Total Applied: \$5,066.22

Apply	Inv. Date	Inv. #	Invoice Amount	Prev. Payments	Balance	Amount Applied
✓	5/18/2022	2	\$1,653.73	\$0.00	\$1,653.73	\$1,653.73
✓	12/28/2022	5	\$3,251.61	\$0.00	\$3,251.61	\$3,251.61
✓	1/4/2023	8	\$34.63	\$0.00	\$34.63	\$34.63
✓	1/19/2023	9	\$26.25	\$0.00	\$26.25	\$26.25
	1/17/2023	10	\$262.50	\$0.00	\$262.50	\$100.00
	1/5/2023	11	\$445.20	\$0.00	\$445.20	



*Excel For Freelancers*

## How To Create Your Own Invoice Payment System In Excel - FROM SCRATCH



[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 400,000 YouTube subscribers, 15,000,000 video views, 200+ comprehensive training videos, and a thriving 50,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



**Microsoft**<sup>®</sup>  
Most Valuable  
Professional



# OUR COURSES & PRODUCTS



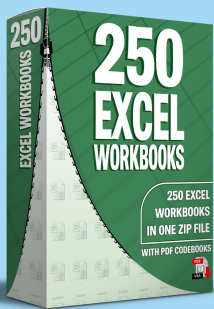
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



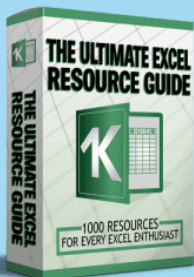
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



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](#)

**Table of Contents**

Projects.....	3
VBAProject.....	3
Documents.....	3
Admin.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub ].....	3
Customers.....	4
(Declarations).....	4
Invoice.....	5
(Declarations).....	5
Worksheet_Change [Sub ].....	5
Worksheet_SelectionChange [Sub ].....	6
InvoiceItems.....	10
(Declarations).....	10
InvoiceList.....	11
(Declarations).....	11
Items.....	12
(Declarations).....	12
PayItems.....	13
(Declarations).....	13
PaymentList.....	14
(Declarations).....	14
Payments.....	15
(Declarations).....	15
Worksheet_Change [Sub ].....	15
Worksheet_SelectionChange [Sub ].....	15
Sheet1.....	16
(Declarations).....	16
ThisWorkbook.....	17
(Declarations).....	17
Forms.....	18
AddCustForm.....	18
(Declarations).....	18
CancelBtn_Click [Sub ].....	18
SaveBtn_Click [Sub ].....	18
Modules.....	19
Admin_Menu_Macros.....	19
(Declarations).....	19
AppEvents_Start [Sub ].....	19
AppEvents_Stop [Sub ].....	19
Company_AddLogo [Sub ].....	19
EditMode [Sub ].....	19
Invoice_Macros.....	20
(Declarations).....	20
Customer_AddUpdate [Sub ].....	20
Invoice_Delete [Sub ].....	20
Invoice_Load [Sub ].....	20
Invoice_New [Sub ].....	21
Invoice_NewPage [Sub ].....	21
Invoice_NextInvoice [Sub ].....	21
Invoice_NextPage [Sub ].....	22
Invoice_PageLoad [Sub ].....	22
Invoice_PrevInvoice [Sub ].....	22
Invoice_PrevPage [Sub ].....	23
Invoice_SavedMsg [Sub ].....	23
Invoice_SaveUpdate [Sub ].....	23
Payments_Macros.....	25
(Declarations).....	25
Payments_AddNew [Sub ].....	25
Payments_Delete [Sub ].....	25
Payments_Load [Sub ].....	25
Payments_LoadOpenInvoices [Sub ].....	26
Payments_Next [Sub ].....	26
Payments_Previous [Sub ].....	27
Payments_SaveUpdate [Sub ].....	27
PopUp_Calendar.....	29

## Table of Contents

(Declarations) .....	29
CalCol [Sub ] .....	29
CalendarHide [Sub ] .....	29
CalendarShow [Sub ] .....	29
CalFormulaReplacement [Sub ] .....	30
CheckForSheet [Sub ] .....	30
CreateCalSht [Sub ] .....	31
GroupCal [Sub ] .....	32
MacroLinkRemover [Sub ] .....	33
NextMonth [Sub ] .....	34
NextYear [Sub ] .....	34
PrevMonth [Sub ] .....	34
PrevYear [Sub ] .....	35
ReplaceCalendar [Sub ] .....	35
SelectDay [Sub ] .....	35
ShowSettings [Sub ] .....	35
UnGroupCal [Sub ] .....	35
Print_Macros .....	36
(Declarations) .....	36
Invoice_Print [Sub ] .....	36
Invoice_SaveAsPDF [Sub ] .....	36

```
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 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                        MsgBox "Please select a correct item from the drop down list"
37                        Exit Sub
38                    End If
39                    Range("L" & Target.Row).Value = Items.Range("B" & ItemDBRow).Value
40                    'Description
41                    Range("M" & Target.Row).Value = Items.Range("C" & ItemDBRow).Value 'Default
42                    Qty
43                    Range("N" & Target.Row).Value = Items.Range("D" & ItemDBRow).Value
44                    'Amount/Price
45                    Range("P" & Target.Row).Value = Items.Range("E" & ItemDBRow).Value 'Taxable
46                Else 'Empty Clear Row
47                    Range("L" & Target.Row & ":N" & Target.Row).ClearContents 'Clear Item
48                    Details
49                    Range("P" & Target.Row).ClearContents 'Clear Tax
50                End If
51            End If
52
53            'On Any Invoice Item Detail Change, Save To Database (not on Invoice Load)
54            If Not Intersect(Target, Range("J10:P44" )) Is Nothing And Range("B1" ).Value =
55                False Then
56                SelRow = Target.Row 'Set Selected Row
57                'Update Invoice Items DB
58                If Range("Q" & SelRow).Value = Empty Then 'New Invoice Item
59                    InvItemRow = InvoiceItems.Range("A9999" ).End(xlUp).Row + 1 'First Avail.

```

1 2 3



```

1 2 3
Row
53 InvoiceItems.Range("A" & InvItemRow).Value = Range("B4").Value 'Invoice
Number
54 InvoiceItems.Range("B" & InvItemRow).Value = Range("B11").Value 'Page
Number
55 InvoiceItems.Range("J" & InvItemRow).Value = SelRow 'Set Invoice Row
56 InvoiceItems.Range("K" & InvItemRow).Value = "=row()" 'Set Row Formula
57 Range("Q" & Target.Row).Value = InvItemRow 'Set Item Row
58 Else 'Existing Invoice Item
59     InvItemRow = Range("Q" & Target.Row).Value 'Existing Item Row
60 End If
61 InvoiceItems.Cells(InvItemRow, Target.Column - 7).Value = Target.Value
62 InvoiceItems.Range("H" & InvItemRow).Value = Range("O" & SelRow).Value 'Update
Total
63 End If
64
65 'On Any Terms Or Inv. Date Change, Update Due Date (not on Invoice Load) (if
applicable)
66 If Not Intersect(Target, Range("E4,E6" )) Is Nothing And Range("E4").Value <> Empty
And Range("E6").Value <> Empty And Range("B1").Value = False Then
67     On Error Resume Next
68     TermRow = Admin.Range("Terms").Find(Range("E6").Value, , xlValues, xlWhole).Row
69
70     On Error GoTo 0
71     If TermRow = 0 Then Exit Sub 'Exit if not found
72     Range("E8").Value = Range("E4").Value + Admin.Range("I" & TermRow).Value
'Update Due Date
73 End If
74
75 '''Customize Invoice Changes
76 If Range("B10").Value = False Then Exit Sub 'Exit unless in Edit mode
77 Dim ItemLblShp As Shape
78 'On Change of Show or Label
79 If Not Intersect(Target, Range("F13:G22" )) Is Nothing And Range("B8").Value =
False And Range("E" & Target.Row).Value <> Empty Then
80     ItemTypeRow = Range("B9").Value 'Selected Item Type Row
81     ItemCol = 22 + ((ItemTypeRow - 13) * 4) + Target.Column 'Column to Save To
82     Cells(Target.Row, ItemCol).Value = Target.Value 'Update Data table
83     'On Change of Label, Show/Hide Label (but not on Item Detail Columns)
84     If Not Intersect(Target, Range("G13:G22" )) Is Nothing And Range("B9").Value <>
16 Then
85         On Error Resume Next
86         Set ItemLblShp = Shapes(Range("E" & Target.Row).Value & "_Lbl" )
87         If Target.Value = Empty Then ItemLblShp.Visible = msoFalse Else ItemLblShp.
Visible = msoCTrue 'Hide/Show Label based on Label Value
88     End If
89 End If
90
91 End Sub
92
93 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
94     Dim ItemNm As String
95     Dim ItemShp As Shape, ItemLblShp As Shape, GrpShp As Shape, Cal As Shape
96     If Target.CountLarge > 2 Then Exit Sub
97     On Error Resume Next
98     Set Cal = Shapes("Calendar" )
99     On Error GoTo 0
100     If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
101     If Shapes("Calendar").Visible = True Then CalendarHide 'Run macro to hide calendar

```

```

1
102
103 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
104     CheckForSheet
105     CalendarShow
106 Else:
107     CheckForSheet
108     CalendarHide
109 End If
110
111 'On Tax Selection Change but not on Invoice Load
112 On Error Resume Next
113 If Not Intersect(Target, Range("P10:P44" )) Is Nothing And Range("B1" ).Value =
    False Then
114     If Target.Value = Chr(252) Then Target.ClearContents Else Target.Value = Chr(252)
115     Range("P8" ).Select 'Select another cell
116 End If
117
118
119 '''Customize Invoice Changes
120
121 If Range("B10" ).Value = False Then Exit Sub 'Exit unless in Edit mode
122 'On Edit Invoice Item Menu Select
123 If Not Intersect(Target, Range("D13:D16" )) Is Nothing Then
124     Range("B8" ).Value = True 'Set Menu Change to True
125     Range("B9" ).Value = Target.Row 'Set Type /Trigger CF
126     Range("E13:G22" ).ClearContents 'Clear Fields
127     ItemCol = 27 + ((Target.Row - 13) * 4)
128     Range("E13:G22" ).Value = Range(Cells(13, ItemCol), Cells(22, ItemCol + 2)).Value
    'Load Item Details
129     Range("B8" ).Value = False 'Set Menu Change To False
130 End If
131
132 'On Show/Hide Selection
133 If Not Intersect(Target, Range("F13:F22" )) Is Nothing And Range("E" & Target.Row).
    Value <> Empty Then
134     Dim InvWidth As Double, DescWidth As Double
135     Set ItemShp = Nothing
136     Set ItemLblShp = Nothing
137     If Target.Value = Chr(252) Then Target.ClearContents Else Target.Value = Chr(252)
138
139     ItemTypeRow = Range("B9" ).Value 'Set Item Type Row
140     ItemRow = Target.Row
141     'If Item Details are selected, then Hide/Show Columns
142     If ItemTypeRow = 16 Then 'Item Details (Columns)
143         ItemCol = Target.Row - 3 'Set Item Column
144         If Target.Value = "" Then 'Hide Column
145             Cells(1, ItemCol).EntireColumn.Hidden = True
146         Else 'Show Column
147             Cells(1, ItemCol).EntireColumn.Hidden = False
148         End If
149         'Adjust Item Description column if its not hidden
150         If ItemCol <> 12 Then
151             InvWidth = Columns("J" ).ColumnWidth + Columns("K" ).ColumnWidth + Columns(
                "L" ).ColumnWidth + Columns("M" ).ColumnWidth + _
152             Columns("N" ).ColumnWidth + Columns("O" ).ColumnWidth + Columns("P" ).
                ColumnWidth
153             DescWidth = Columns("L" ).ColumnWidth
            If InvWidth > 546.5 Then Columns("L" ).ColumnWidth = (DescWidth - (InvWidth
                - 93.6))

```

```

154     If InvWidth < 546.5 Then Columns("L" ).ColumnWidth = (DescWidth + (93.6 -
155         InvWidth))
156     End If
157     Else 'Invoice Fields (Shapes)
158         ItemCol = 29 + ((ItemTypeRow - 13) * 4)
159         ItemNm = Range("E" & Target.Row).Value 'Set Item Name
160
161         'Check For Item Value Shape
162         On Error Resume Next
163         Set ItemShp = Shapes(ItemNm)
164         On Error GoTo 0
165         If ItemShp Is Nothing Then 'Create & Position Value Shape if it does not
166             exist
167             If ItemNm <> "Logo" Then
168                 Shapes("SampleInvShp" ).Duplicate.Name = ItemNm 'Create Shape If
169                 missing
170             Else 'Logo
171                 If Admin.Range("C10" ).Value = Empty Or Dir(Admin.Range("C10" ).Value,
172                 vbDirectory) = "" Then GoTo CheckForLbl
173                 On Error Resume Next
174                 Pictures.Insert(Admin.Range("C10" ).Value).Name = ItemNm 'Set Logo
175                 Picture
176                 On Error GoTo 0
177             End If
178             Set ItemShp = Shapes(ItemNm)
179             With ItemShp
180                 If ItemNm <> "Logo" Then .DrawingObject.Formula = "=" & Cells(ItemRow,
181                 ItemCol + 1).Address
182                 .Left = Range("L1" ).Left
183                 .Top = Range("L5" ).Top
184                 .Placement = xlMove 'Move but not size with cells (columns)
185             End With
186             End If
187             CheckForLbl:
188             'Check For Item Label Shape
189             On Error Resume Next
190             Set ItemLblShp = Shapes(ItemNm & "_Lbl" )
191             On Error GoTo 0
192             If ItemLblShp Is Nothing Then 'Create & Position Value Shape if it does not
193             exist
194             Shapes("SampleInvShp" ).Duplicate.Name = ItemNm & "_Lbl" 'Create Shape If
195             missing
196             Set ItemLblShp = Shapes(ItemNm & "_Lbl" )
197             With ItemLblShp
198                 .DrawingObject.Formula = "=" & Cells(ItemRow, ItemCol).Address
199                 .Left = Range("L1" ).Left - 50
200                 .Top = Range("L5" ).Top
201                 .Placement = xlMove
202             End With
203             End If
204             'Group Shapes if group does not exist
205             On Error Resume Next
206             Set GrpShp = Shapes(ItemNm & "_Grp" )
207             On Error GoTo 0
208             If GrpShp Is Nothing Then 'Group Shapes
209             Shapes.Range(Array(ItemShp.Name, ItemLblShp.Name)).Group.Name = ItemNm &
210             "_Grp"
211             Shapes(ItemNm & "_Grp" ).Placement = xlMove 'Set to Move but not size with
212             cells

```

```
1 2 3 4
205 End If
206
207 If Target.Value = Empty Or Target.Offset(0, 1).Value = Empty Then 'Ungroup
    and Hide on Empty Label
    On Error Resume Next
    Shapes(ItemNm & "_Grp" ).Ungroup
    On Error Resume Next
    ItemLblShp.Visible = msoFalse
208 Else
209     ItemLblShp.Visible = msoCTrue 'Show/Hide Invoice Label
210 End If
211 If Target.Value = Empty Then 'Hidden
212     ItemShp.Visible = msoFalse
213 Else 'Visible
214     ItemShp.Visible = msoCTrue 'Show/Hide Invoice Field Value
215 End If
216 End If
217 Range("F2" ).Select
218 End If
219 End Sub
```

1 Option Explicit

2

1 Option Explicit

2

1

Option Explicit

1 Option Explicit

2



1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     'On Customer Change for new payments, not on customer load,
5     If Not Intersect(Target, Range("F3" )) Is Nothing And Range("B2" ).Value = False _
6     And Range("B4" ).Value = Empty And Range("F3" ).Value <> "" Then
7         Payments_LoadOpenInvoices
8
9     'On Applied Payment Amount change but not on payment load then select/Unselect
10    apply
11    If Not Intersect(Target, Range("J11:J99" )) Is Nothing And Range("E" & Target.Row).
12    Value <> Empty And Range("B2" ).Value = False Then
13        If Target.Value = Empty Then Range("D" & Target.Row).ClearContents Else Range(
14        "D" & Target.Row).Value = Chr(252)
15    End If
16 End Sub
17
18 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
19     If Target.CountLarge > 1 Then Exit Sub
20     If Not Intersect(Target, Range("D11:D999" )) Is Nothing And Range("E" & Target.Row)
21     .Value <> Empty Then
22         If Target.Value = "" Then Target.Value = Chr(252) Else Target.ClearContents
23         'Check For Applied Above Payment Amount
24         If Target.Value = Chr(252) Then 'Payment applied
25             Range("J" & Target.Row).Value = Range("I" & Target.Row).Value 'Set Amt.
26             Applied = Balance
27         Else 'Clear Applied Payment
28             Range("J" & Target.Row).ClearContents 'Clear Applied Amount
29         End If
30         Range("F3" ).Select
31     End If
32 End Sub
```

1  
2

Option Explicit

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 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     With Invoice
34         If .Shapes("EditModeBack" ).TextFrame2.TextRange.Text = "Off" Then 'Turn On
35             .Shapes("EditModeSwitch" ).Fill.ForeColor.RGB = RGB(0, 176, 80) 'Turn To
36                 Green Color
37             .Shapes("EditModeBack" ).TextFrame2.TextRange.Text = "On"
38             MovInc = 1
39             .Range("B10" ).Value = True 'Set Edit Mode To True
40         Else 'Turn Off
41             MovInc = -1
42             .Shapes("EditModeSwitch" ).Fill.ForeColor.RGB = RGB(127, 125, 127) 'Turn to
43                 gray color
44             .Shapes("EditModeBack" ).TextFrame2.TextRange.Text = "Off"
45             .Range("B10" ).Value = False 'Set Edit Mode To False
46         End If
47         Application.ScreenUpdating = False
48         For BtnMov = 1 To 29
49             .Shapes("EditModeSwitch" ).Left = .Shapes("EditModeSwitch" ).Left + MovInc
50         Next BtnMov
51         Application.ScreenUpdating = True
52     End With
53 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()
1

```

```

1
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
        'Fill Invoice Details
67   Next FieldRow
68   .Range("B12").Value = InvoiceList.Range("I" & InvRow).Value 'Set Total Pages
69   Invoice_PageLoad 'Load Page Items
70   .Range("B1").Value = False 'Set inv. Load to false
71   AppEvents_Start
72 End With
73 End Sub
74
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
        'Default Payment Term Row
85     On Error GoTo 0
86     If TermRow <> 0 Then .Range("E6").Value = Admin.Range("H" & TermRow).Value
        'Set Default Term
87     On Error Resume Next
88     StatusRow = Admin.Range("F6:F12").Find(Chr(252), , xlValues, xlWhole).Row
        'Default Status Row
89     On Error GoTo 0
90     If StatusRow <> 0 Then .Range("E7").Value = Admin.Range("E" & StatusRow).Value
        'Set Default Status
91     .Range("E5").Select
92   End With
93 End Sub
94
95 Sub Invoice_NewPage()
96   Invoice_SaveUpdate 'Save New Last page limit
97   Invoice.Range("B11").Value = Invoice.Range("B12").Value + 1 'Set Selected page to
        new page
98   Invoice.Range("B12").Value = Invoice.Range("B12").Value + 1 ' Increment page to +
        1
99   Invoice_PageLoad
100  Invoice_SaveUpdate 'Save New Last page limit
101 End Sub
102
103 Sub Invoice_NextInvoice()
104   With Invoice
105     Dim MaxInvNumb As Long
106     On Error Resume Next
107     MaxInvNumb = Application.WorksheetFunction.Max(InvoiceList.Range("Invoice_ID"))
108     On Error GoTo 0

```

1 2



```

1 2
109     InvNumb = .Range("B4" ).Value
110     If MaxInvNumb = 0 Then
111         MsgBox "Please create and save an Invoice first"
112         Exit Sub
113     End If
114     If InvNumb = MaxInvNumb Then
115         MsgBox "You are at the last Invoice"
116         Exit Sub
117     End If
118
119
120     If InvNumb = 0 Or .Range("B5" ).Value = Empty Then 'On New Invoice
121         InvRow = 3 'On Empty Invoice Go to First one created
122     Else 'On Existing Inv. find Next one
123         InvRow = InvoiceList.Range("Invoice_ID" ).Find(InvNumb, , xlValues, xlWhole).
124         Row + 1
125     End If
126     .Range("B4" ).Value = InvoiceList.Range("A" & InvRow).Value
127     Invoice_Load
128 End With
129 End Sub
130
131 Sub Invoice_NextPage()
132     If Invoice.Range("B11" ).Value = Invoice.Range("B12" ).Value Then
133         MsgBox "You are on the last page"
134         Exit Sub
135     End If
136     Invoice.Range("B11" ).Value = Invoice.Range("B11" ).Value + 1 'Increment Page # Up
137     Invoice_PageLoad
138 End Sub
139
140 Sub Invoice_PageLoad()
141     If Invoice.Range("B5" ).Value = Empty Then Exit Sub 'Exit on no invoice
142     Invoice.Range("B1" ).Value = True 'Set Inv. Load to true
143     Invoice.Range("J10:N44,P10:Q44" ).ClearContents 'Clear Existing Page Items
144     With InvoiceItems
145         LastRow = .Range("A99999" ).End(xlUp).Row
146         If LastRow < 4 Then GoTo NoItems
147         .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
148             "M2:N3" ), CopyToRange:=.Range("P2:W2" ), Unique:=True
149         LastResultRow = .Range("P99999" ).End(xlUp).Row
150         If LastResultRow < 3 Then GoTo NoItems
151         For ResultRow = 3 To LastResultRow
152             InvItemRow = .Range("P" & ResultRow).Value 'Get Invoice Row
153             Invoice.Range("J" & InvItemRow & ":N" & InvItemRow).Value = .Range("Q" &
154                 ResultRow & ":U" & ResultRow).Value 'Item Details
155             Invoice.Range("P" & InvItemRow & ":Q" & InvItemRow).Value = .Range("V" &
156                 ResultRow & ":W" & ResultRow).Value 'Tax & DB Row
157         Next ResultRow
158     NoItems:
159 End With
160 Invoice.Range("B1" ).Value = False 'Set inv. Load to false
161 End Sub
162
163 Sub Invoice_PrevInvoice()
164     With Invoice
165         Dim MinInvNumb As Long
166         On Error Resume Next
167         MinInvNumb = Application.WorksheetFunction.Min(InvoiceList.Range("Invoice_ID" ))
168         On Error GoTo 0

```

```

1 2
166 If MinInvNumb = 0 Then
167     MsgBox "Please create and save an Invoice first"
168     Exit Sub
169 End If
170 InvNumb = .Range("B4" ).Value
171 If InvNumb = 0 Or .Range("B5" ).Value = Empty Then 'On New Invoice
172     InvRow = InvoiceList.Range("A99999" ).End(xlUp).Row 'On Empty Invoice Go to
        last one created
173 Else 'On Existing Inv. find Previous one
174     InvRow = InvoiceList.Range("Invoice_ID" ).Find(InvNumb, , xlValues, xlWhole).
        Row - 1
175 End If
176 If .Range("B6" ).Value = 1 Or MinInvNumb = 0 Or MinInvNumb = .Range("B4" ).Value
    Then
177     MsgBox "You are at the first invoice"
178     Exit Sub
179 End If
180 .Range("B4" ).Value = InvoiceList.Range("A" & InvRow).Value 'Place Inv. ID
    inside cell
181 Invoice_Load
182 End With
183 End Sub
184
185 Sub Invoice_PrevPage()
186 If Invoice.Range("B11" ).Value = 1 Then
187     MsgBox "You are on the first page"
188     Exit Sub
189 End If
190 Invoice.Range("B11" ).Value = Invoice.Range("B11" ).Value - 1 'Increment Page #
    Down
191 Invoice_PageLoad
192 End Sub
193 Sub Invoice_SavedMsg()
194 With Invoice.Shapes("InvSavedMsg" )
195     Dim i As Long, Delay As Double, StartTime As Double
196     .Visible = msoCTrue
197     For i = 1 To 100
198         .Fill.Transparency = i / 100
199         Delay = 0.009
200         StartTime = Timer
201         Do
202             DoEvents
203         Loop While Timer - StartTime < Delay
204     Next i
205     .Visible = msoFalse
206 End With
207 End Sub
208 Sub Invoice_SaveUpdate()
209 With Invoice
210     If .Range("B3" ).Value = Empty Then
211         MsgBox "Please make sure to add a customer before saving invoice"
212         Exit Sub
213     End If
214     AppEvents_Stop
215     'Deterermine New Invoice/Existing Invoice
216     If .Range("B5" ).Value = Empty Then 'new Invoice
217         InvRow = InvoiceList.Range("A99999" ).End(xlUp).Row + 1 'First Avail Row
218         .Range("B4" ).Value = .Range("B6" ).Value 'Next Inv. #
219         InvoiceList.Range("A" & InvRow).Value = .Range("B4" ).Value 'Next Inv. #
220         InvoiceList.Range("J" & InvRow).Value = "=H" & InvRow &
            "-IFERROR(SUMIF(PayItem_InvID,A" & InvRow & ",PayItem_Amount),0)"

```

1 2 3

```
1 2 3
221 Else 'Existing Invoice
222     InvRow = .Range("B5").Value 'Invoice Row
223 End If
224 For FieldRow = 4 To 9
225     InvoiceList.Cells(InvRow, FieldRow - 2).Value = .Range("E" & FieldRow).Value
        'Add Invoice Details
226 Next FieldRow
227     InvoiceList.Range("H" & InvRow).Value = .Range("AD21").Value 'Invoice Total
228     InvoiceList.Range("I" & InvRow).Value = .Range("B12").Value 'Total Pages
229 End With
230 AppEvents_Start
231 If Application.Caller = "SaveBtn" Then Invoice_SavedMsg 'Run Invoice Fade Out
        Message
232 End Sub
```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, PayRow As Long, PayCol As Long
3 Dim ResultRow As Long, LastPayItemRow As Long, PayItemRow As Long, PayItemDBRow As Long
4
5 Sub Payments_AddNew()
6     Payments.Range("B3,F3:G3,J3,F5:G5,J5,F7:J8,D11:K9999").ClearContents 'Clear Fields
7     Payments.Range("J3").Value = Date 'Set Default Date
8 End Sub
9
10 Sub Payments_Delete()
11     If MsgBox("Are you sure you want to delete this payment?" , vbYesNo, "Delete
12     Payment" ) = vbNo Then Exit Sub
13     With Payments
14         If .Range("B4").Value = Empty Then GoTo NotSaved
15         PayRow = .Range("B4").Value 'Pay Row
16         PaymentList.Range(PayRow & ":" & PayRow).EntireRow.Delete 'Delete Payment Row
17     With PayItems
18         LastRow = .Range("A9999").End(xlUp).Row
19         If LastRow < 4 Then GoTo NotSaved
20         .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
21         "J2:J3" ), CopyToRange:=.Range("O3:T3" ), Unique:=True
22         LastResultRow = .Range("O9999").End(xlUp).Row
23         If LastResultRow < 4 Then GoTo NotSaved
24         If LastResultRow < 5 Then GoTo SkipSort
25     With .Sort
26         .SortFields.Clear
27         .SortFields.Add Key:=PayItems.Range("T4" ), SortOn:=xlSortOnValues, Order:=
28         xlDescending, DataOption:=xlSortNormal 'Sort
29         .SetRange PayItems.Range("O4:T" & LastResultRow) 'Set Range
30         .Apply 'Apply Sort
31     End With
32     SkipSort:
33     On Error Resume Next
34     For ResultRow = 4 To LastResultRow
35         PayItemDBRow = .Range("T" & ResultRow).Value 'Pay Item DB Row
36         .Range(PayItemDBRow & ":" & PayItemDBRow).EntireRow.Delete 'Delete Pay
37         Item DB Row
38     Next ResultRow
39 End With
40 NotSaved:
41 Payments_AddNew
42 End With
43 End Sub
44
45 Sub Payments_Load()
46     With Payments
47         If .Range("B4").Value = Empty Then
48             MsgBox "Please make sure to select a correct payment"
49             Exit Sub
50         End If
51         PayRow = .Range("B4").Value 'Payment Row
52         .Range("B2").Value = True 'Payment Load To True
53         .Range("F3:G3,J3,F5:G5,J5,F7:J8,D11:K9999").ClearContents
54         For PayCol = 2 To 6
55             .Range(PaymentList.Cells(1, PayCol).Value).Value = PaymentList.Cells(PayRow,
56             PayCol).Value 'Load From DB
57         Next PayCol
58     End With
59 End Sub

```

1 2

```

55 | 1 2
56 | 'Load Pay Items
57 | With PayItems
58 |     .Range("M4:T999").ClearContents 'Clear Any Previous Result
59 |     LastRow = .Range("A99999").End(xlUp).Row
60 |     If LastRow < 4 Then GoTo NoData
61 |     .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
62 |         "J2:J3" ), CopyToRange:=.Range("O3:T3" ), Unique:=True
63 |     LastResultRow = .Range("O99999").End(xlUp).Row
64 |     If LastResultRow < 4 Then GoTo NoData
65 |     .Range("M4:N" & LastResultRow).Formula = .Range("M1:N1").Formula 'Bring
66 |     Apply and Inv. Date Formulas
67 |     .Range("P4:R" & LastResultRow).Formula = .Range("P1:R1").Formula ' Inv.
68 |     Amount, Prev. Payments, Balance
69 |     Payments.Range("D11:K" & LastResultRow + 7).Value = .Range("M4:T" &
70 |     LastResultRow).Value 'Bring over pay items
71 |     NoData:
72 |     Payments.Range("B2").Value = False 'Payment Load To False
73 | End With
74 | End With
75 | End Sub
76 |
77 | Sub Payments_LoadOpenInvoices()
78 |     Payments.Range("D11:K9999").ClearContents 'Clear Any Previous Data
79 |
80 |     With Invoicelist
81 |         LastRow = .Range("A99999").End(xlUp).Row 'Last Row
82 |         If LastRow < 4 Then Exit Sub
83 |         .Range("A2:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
84 |             "L2:M3" ), CopyToRange:=.Range("P2:T2" ), Unique:=True
85 |         LastResultRow = .Range("P99999").End(xlUp).Row
86 |         If LastResultRow < 3 Then Exit Sub
87 |         Payments.Range("B2").Value = True 'Set Payment Load to true
88 |         .Range("S3:S" & LastResultRow).Formula = .Range("S1").Formula 'Total Payments
89 |         Formula
90 |         Payments.Range("E11:I" & LastResultRow + 8).Value = .Range("P3:T" &
91 |         LastResultRow).Value
92 |     End With
93 |     Payments.Range("B2").Value = False 'Set Payment Load to False
94 | End Sub
95 |
96 | Sub Payments_Next()
97 |     Dim MaxPayID As Long, PayID As Long
98 |     With Payments
99 |         On Error Resume Next
100 |         MaxPayID = Application.WorksheetFunction.Max(PaymentList.Range("Pay_ID" ))
101 |         On Error GoTo 0
102 |         If MaxPayID = 0 Then
103 |             MsgBox "Please create a payment first"
104 |             Exit Sub
105 |         End If
106 |         PayID = .Range("B3").Value 'Payment ID
107 |
108 |         If PayID = 0 Or .Range("B4").Value = Empty Then 'Load Last Payment Created
109 |             PayRow = 4 'On new Payment, Go first one created
110 |         Else
111 |             PayRow = .Range("B4").Value + 1 'Pay Row + 1
112 |         End If
113 |
114 |         If MaxPayID = PayID Then 'Last Payment

```

1 2 3

```

1 2 3
109     MsgBox "You are on the last payment"
110     Exit Sub
111     End If
112
113     .Range("B3" ).Value = PaymentList.Range("A" & PayRow).Value 'Set Payment ID
114     Payments_Load 'Load Payment
115     End With
116     End Sub
117
118     Sub Payments_Previous()
119         Dim MinPayID As Long, PayID As Long
120         With Payments
121
122             On Error Resume Next
123             MinPayID = Application.WorksheetFunction.Min(PaymentList.Range("Pay_ID" ))
124             On Error GoTo 0
125             If MinPayID = 0 Then
126                 MsgBox "Please create a payment first"
127                 Exit Sub
128             End If
129             PayID = .Range("B3" ).Value 'Payment ID
130             If PayID = 0 Or .Range("B4" ).Value = Empty Then 'Load Last Payment Created
131                 PayRow = PaymentList.Range("A9999" ).End(xlUp).Row 'Last Row
132             Else
133                 PayRow = .Range("B4" ).Value - 1 'Pay Row -1
134             End If
135
136             If PayRow = 3 Or MinPayID = .Range("B3" ).Value Then 'First Payment
137                 MsgBox "You are on the first payment"
138                 Exit Sub
139             End If
140
141             .Range("B3" ).Value = PaymentList.Range("A" & PayRow).Value 'Set Payment ID
142             Payments_Load 'Load Payment
143         End With
144     End Sub
145
146     Sub Payments_SaveUpdate()
147         With Payments
148             'Check For Required Fields
149             If .Range("F3" ).Value = Empty Or .Range("J3" ).Value = Empty Or .Range("J5" ).
150                 Value = Empty Then
151                 MsgBox "Please make sure to add in a Customer, Payment Date and Payment
152                     Amount before saving"
153                 Exit Sub
154             End If
155
156             'Check To Make sure Payment Amount = Applied amount
157             If .Range("J5" ).Value <> .Range("J9" ).Value Then
158                 MsgBox "Please make sure Payment Amount is equal to Applied Amount"
159                 Exit Sub
160             End If
161
162             If .Range("B4" ).Value = Empty Then 'New Payment
163                 PayRow = PaymentList.Range("A99999" ).End(xlUp).Row + 1 'First Avail Row
164                 .Range("B3" ).Value = .Range("B5" ).Value 'Next payment ID
165                 PaymentList.Range("A" & PayRow).Value = .Range("B3" ).Value 'Pay ID
166             Else 'Existing Payment
167                 PayRow = .Range("B4" ).Value
168             End If
169             For PayCol = 2 To 6

```

```
168      1 2 3      PaymentList.Cells(PayRow, PayCol).Value = .Range(PaymentList.Cells(1, PayCol).
      Value).Value 'Save Form Data
169      Next PayCol
170
171      'Save Pay Items
172      LastPayItemRow = .Range("E99999" ).End(xlUp).Row 'Last Pay Item
173      For PayItemRow = 11 To LastPayItemRow
174          If .Range("D" & PayItemRow).Value = Chr(252) Then 'Selected Row
175              If .Range("K" & PayItemRow).Value = Empty Then 'New Pay Item Row
176                  PayItemDBRow = PayItems.Range("A99999" ).End(xlUp).Row + 1 'First Avail
                  Pay Item Row
177                  PayItems.Range("A" & PayItemDBRow).Value = .Range("B3" ).Value
                  'Payment ID
178                  PayItems.Range("F" & PayItemDBRow).Value = "=Row()" 'Item Row Formula
179                  .Range("K" & PayItemRow).Value = PayItemDBRow 'Database Row
180              Else 'Previously Saved
181                  PayItemDBRow = .Range("K" & PayItemRow).Value 'Existing Pay Item Row
182              End If
183              PayItems.Range("B" & PayItemDBRow).Value = .Range("F" & PayItemRow).Value
                  'Invoice ID
184              PayItems.Range("C" & PayItemDBRow).Value = .Range("F3" ).Value 'Customer
185              PayItems.Range("D" & PayItemDBRow).Value = .Range("J3" ).Value 'Pay Date
186              PayItems.Range("E" & PayItemDBRow).Value = .Range("J" & PayItemRow).Value
                  'Amount
187          End If
188      Next PayItemRow
189  End With
190 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
32         CalPopUp Sheet
33 End Sub
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" ).
55         Value) 'Set Month
56         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
57         Value) 'Set Year
58         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
59         ' UnGroupCal
60         If InStr(.Shapes("1Day" ).DrawingObject.Formula, "]" ) <> 0 Then 'Run Workbook

```

1 2 3



```

1 2 3
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",
72     "CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible = False '
73     .Shapes("Calendar").Left = SelCell.Left
74     .Shapes("Calendar").Placement = xlMove
75     .Shapes("Calendar").Top = SelCell.Offset(1, 0).Top
76     If Sheets("CalPopUp").Range("A6").Value > 0 Then
77         .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
78         "42Day")).Visible = True
79     Else:
80         .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
81         "42Day")).Visible = False
82     End If
83     Sheets("CalPopUp").Range("A7").Value = SelCell.Address
84     ActiveCell.Select
85 End With
86 Exit Sub
87 NoCal:
88 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
89 calendar over from another sheet and paste into this sheet"
90 End Sub
91
92 Sub CalFormulaReplacement()
93     With ActiveSheet
94         Dim DayNum, ColNum, RowNum As Long
95         Dim Shp As Shape
96         ColNum = 2
97         RowNum = 1
98         For DayNum = 1 To 42
99             .Shapes(DayNum & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(RowNum
100             , ColNum).Address
101             ColNum = ColNum + 1
102             If ColNum = 9 Then
103                 ColNum = 2
104                 RowNum = RowNum + 1
105             End If
106         Next DayNum
107         .Shapes("Month").DrawingObject.Formula = "=CalPopUp!$A$4"
108         .Shapes("Year").DrawingObject.Formula = "=CalPopUp!$A$2"
109     End With
110 End Sub
111
112 Sub CheckForSheet()
113     'Checks for existance of Calendar Pop-up Worksheet
114     Dim ws As Worksheet
115     On Error GoTo CreateWS
116     Set ws = ActiveWorkbook.Sheets("CalPopUp")
117     Exit Sub
118 CreateWS:
119     CreateCalSht

```

```

1
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"
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 2

```

1 2
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$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
177 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
    ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
178 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
    ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
179 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
    ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
180 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
    ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
181 .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
    ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
182 .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
    ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
183 .Range("B2" ).Value = "=H1+1"
184 .Range("C2" ).Value = "=B2+1"
185 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
186 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
187 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$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:
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()

```

1

```

1
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
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

```

1 2

```

1 2
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 stragglng 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     With Sheets("CalPopUp" )
313         If .Range("A3" ).Value = 1 Then
314             .Range("A3" ).Value = 12
315             .Range("A2" ).Value = .Range("A2" ).Value - 1
316         End If
317     End With
318 End Sub
1 2 3

```

```

1 2 3
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 Sub PrevYear()
327     ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
328         "CalPopUp" ).Range("A2" ).Value - 1
329 End Sub
330 Sub ReplaceCalendar() 'Shape Deleted
331     ThisWorkbook.Sheets("CalPopUp" ).Shapes("Calendar" ).Copy 'Copy From Developers
332     sheet
333     ActiveCell.Select 'Select the active Cell
334     ActiveSheet.Paste 'Paste in Calendar
335 End Sub
336 .....
337 ''''Select Day Of The Month
338 .....
339 Sub SelectDay()
340     Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
341     DayNumb = Replace(Application.Caller, "Day" , "" )
342     RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7 , 0)
343     ColNumb = DayNumb Mod 7 + 1
344     If ColNumb = 1 Then ColNumb = 8
345     'On Error Resume Next
346     If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
347     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
348         ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
349     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
350     ActiveCell.Offset(0, 1).Select
351 End Sub
352 Sub ShowSettings()
353     'Show or Hide Calendar Settings Panel
354     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
355         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
356             "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
357             = False
358     Else:
359         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
360             "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
361             = True
362     End If
363 End Sub
364 Sub UnGroupCal()
365     On Error Resume Next
366     ActiveSheet.Shapes("Calendar" ).Ungroup
367     ActiveSheet.Shapes("NextMonth" ).Ungroup
368     ActiveSheet.Shapes("PrevMonth" ).Ungroup
369     On Error GoTo 0
370 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     MsgBox "Please join our Patreon Program to see an updated training and this Save As
17     PDF Feature" & vbCrLf & "https://www.patreon.com/ExcelForFreelancers"
18 End Sub
```

—  
\_, 7, 15, 33

**A**

Activate, 31  
ActiveCell, 29, 30, 35  
ActiveSheet, 29-31, 33-35  
ActiveWorkbook, 30-32  
ActSht, 31  
Add, 19, 20, 25, 31, 32  
AddCustForm, 18, 20  
Address, 8, 30, 31  
Admin, 6, 8, 19, 21  
AdvancedFilter, 20, 22, 25, 26  
After, 31  
AllowMultiSelect, 19  
AppEvents\_Start, 19-21, 24  
AppEvents\_Stop, 19-21, 23  
Application, 5, 18, 19, 21, 22, 24, 26, 27, 29, 35  
Apply, 20, 25  
Array, 8, 29, 30, 33-35  
AutoFill, 31, 32

**B**

Bold, 29, 30, 34  
BtnMov, 19

**C**

Cal, 6  
CalCol, 29, 31  
Calculation, 19  
CalendarHide, 6, 7, 29  
CalendarShow, 7, 29  
CalFormulaReplacement, 30  
Caller, 24, 29, 35  
CancelBtn\_Click, 18  
Cells, 6-8, 18, 20, 21, 24, 25, 28, 30, 31, 35  
CheckForLbl, 8  
CheckForSheet, 7, 30  
Chr, 3, 7, 15, 21, 28, 31, 32  
Clear, 20, 25  
ClearContents, 3, 5, 7, 15, 21, 22, 25, 26  
ColCnt, 31  
ColNum, 30  
ColNumb, 35  
Column, 6  
Columns, 7, 8  
ColumnWidth, 7, 8  
Company\_AddLogo, 19  
Control, 18, 20  
Controls, 18, 20  
Copy, 31, 35  
CopyToRange, 20, 22, 25, 26  
Count, 31  
CountLarge, 3, 5, 6, 15  
CreateCalSht, 30, 31  
CreateWS, 30  
CriteriaRange, 20, 22, 25, 26  
CustCol, 18, 20  
Customer\_AddUpdate, 20  
CustomerFld, 18, 20  
Customers, 18, 20  
CustRow, 18, 20

**D**

DataOption, 20, 25  
Day, 30  
DayCnt, 31

DayName, 29, 30, 34  
DayNum, 29, 30  
DayNumb, 35  
Delay, 23  
Delete, 20, 25  
DescWidth, 7, 8  
Destination, 31, 32  
Dir, 8  
DoEvents, 23  
DrawingObject, 8, 29-31  
Duplicate, 8

**E**

EditMode, 19  
Empty, 3, 5-9, 15, 18, 20-23, 25-29, 34, 35  
EnableEvents, 5, 19  
EntireColumn, 7  
EntireRow, 20, 25  
Explicit, 3-5, 10-20, 25, 36

**F**

Field1, 18  
FieldRow, 20, 21, 24  
FileDialog, 19  
Fill, 19, 23, 29, 30, 34  
Filters, 19  
Find, 5, 6, 21-23  
Font, 29, 30, 34  
ForeColor, 19, 29, 30, 34  
Formula, 8, 26, 29-31

**G**

Group, 8, 33  
GroupCal, 32  
GrpShp, 6, 8

**H**

HeadShp, 20  
Hidden, 7  
Hide, 18

**I**

i, 23  
Insert, 8  
InStr, 29, 34  
Intersect, 3, 5-7, 15  
InvItemRow, 5, 6, 20, 22  
InvNumb, 20, 22, 23  
Invoice, 18-23, 36  
Invoice\_Delete, 20  
Invoice\_Load, 5, 20, 22, 23  
Invoice\_New, 20, 21  
Invoice\_NewPage, 21  
Invoice\_NextInvoice, 21  
Invoice\_NextPage, 22  
Invoice\_PageLoad, 21-23, 36  
Invoice\_PrevInvoice, 22  
Invoice\_PrevPage, 23  
Invoice\_Print, 36  
Invoice\_SaveAsPDF, 36  
Invoice\_SavedMsg, 23, 24  
Invoice\_SaveUpdate, 21, 23, 36  
InvoiceItems, 5, 6, 20, 22  
Invoicelist, 20-24, 26  
InvRow, 20-24  
InvWidth, 7, 8  
IsDate, 29  
ItemCol, 5-8  
ItemDBRow, 5, 20  
ItemLblShp, 6-9

ItemNm, 6, 8, 9  
ItemRow, 5, 7, 8, 20  
Items, 5  
ItemShp, 6-9  
ItemTypeRow, 5-8

**K**

Key, 20, 25

**L**

LastItemRow, 20  
LastPage, 36  
LastPayItemRow, 25, 28  
LastResultRow, 20, 22, 25, 26  
LastRow, 20, 22, 25, 26  
Left, 8, 19, 30, 34  
Len, 34  
LogoPic, 19

**M**

MacroLink, 33, 34  
MacroLinkRemover, 30, 33  
Max, 18, 21, 26  
MaxInvNumb, 21, 22  
MaxPayID, 26  
Min, 22, 27  
MinInvNumb, 22, 23  
MinPayID, 27  
Month, 29, 31  
MovInc, 19  
MsgBox, 5, 18, 20-23, 25-27, 30, 32, 36  
msoCTrue, 6, 9, 23, 30, 33  
msoFalse, 6, 9, 23, 29, 34, 35  
msoFileDialogFilePicker, 19  
msoTrue, 30

**N**

Name, 8, 31-33  
Names, 32  
NewLink, 33, 34  
NextMonth, 34  
NextShp, 33, 34  
NextYear, 34  
NoCal, 29, 30, 32  
NoData, 26  
NoItems, 22  
NoSelection, 19  
NotSaved, 20, 25  
NumberFormat, 32

**O**

Offset, 9, 30, 35  
OnAction, 31, 33, 34  
Order, 20, 25

**P**

PageNumb, 36  
Paste, 31, 35  
PayCol, 25, 27, 28  
PayID, 26, 27  
PayItemDBRow, 25, 28  
PayItemRow, 25, 28  
PayItems, 25, 26, 28  
PaymentList, 25-28  
Payments, 25-27  
Payments\_AddNew, 25  
Payments\_Delete, 25  
Payments\_Load, 25, 27  
Payments\_LoadOpenInvoices, 15, 26  
Payments\_Next, 26



Payments\_Previous, 27  
 Payments\_SaveUpdate, 27  
 PayRow, 25-28  
 PicPath, 19  
 Pictures, 8  
 Placement, 8, 30, 33  
 PrevMonth, 34  
 PrevYear, 35  
 PrintOut, 36

**R**

Range, 3, 5-9, 15, 18-36  
 RefersTo, 32  
 Replace, 35  
 ReplaceCalendar, 6, 35  
 ResultRow, 20, 22, 25  
 RGB, 19, 29, 30, 34  
 Right, 34  
 RoundUp, 35  
 Row, 3, 5-8, 15, 18, 20-23, 25-28  
 RowCnt, 31  
 RowNum, 30  
 RowNumb, 35

**S**

SaveBtn\_Click, 18  
 ScreenUpdating, 19  
 SelCell, 29, 30  
 SelectDay, 35  
 SelectedItems, 19  
 Selection, 29, 33  
 SelRow, 5, 6  
 SetRange, 20, 25  
 Shape, 6, 20, 30, 33  
 ShapeRange, 29, 33  
 Shapes, 6, 8, 9, 19, 23, 29-31, 33-35  
 Sheets, 29-31, 34, 35  
 Show, 19, 20  
 ShowSettings, 35  
 Shp, 30, 33, 34  
 SingleRow, 20  
 SkipSort, 25  
 Sort, 20, 25  
 SortFields, 20, 25  
 SortOn, 20, 25  
 Split, 34  
 SplitLink, 33, 34  
 StartTime, 23  
 StatusRow, 20, 21

**T**

Target, 3, 5-9, 15  
 TermRow, 5, 6, 20, 21  
 Text, 19  
 TextFrame2, 19, 29, 30, 34  
 TextRange, 19, 29, 30, 34  
 ThisWorkbook, 31, 34, 35  
 Timer, 23  
 Title, 19  
 Top, 8, 30  
 Transparency, 23

**U**

Ungroup, 9, 35  
 UnGroupCal, 31, 35  
 Unique, 20, 22, 25, 26  
 Unload, 20

**V**

Value, 3, 5-9, 15, 18-32, 34-36

vbCrLf, 36  
 vbDirectory, 8  
 vbNo, 20, 25  
 vbYesNo, 20, 25  
 Visible, 6, 9, 23, 29-31, 33-35

**W**

Worksheet, 30, 31  
 Worksheet\_Change, 5, 15  
 Worksheet\_SelectionChange, 3, 6, 15  
 WorksheetFunction, 18, 21, 22, 26, 27, 35  
 ws, 30, 31

**X**

xlCalculationAutomatic, 19  
 xlCalculationManual, 19  
 xlDescending, 20, 25  
 xlFillDefault, 31, 32  
 xlFilterCopy, 20, 22, 25, 26  
 xlMove, 8, 30, 33  
 xlSheetHidden, 31  
 xlSortNormal, 20, 25  
 xlSortOnValues, 20, 25  
 xlUp, 5, 18, 20, 22, 23, 25-28  
 xlValues, 5, 6, 21-23  
 xlWhole, 5, 6, 21-23

**Y**

Year, 29, 31

# 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*