

VBA SOURCE CODE BOOK

PAYMENT REMINDER WITH PENALTIES - UPDATED

REMINDERS

Reminder Name	Status	Variables
15 Days Notice	Active	[CustName]
30 Day Notice		[InvNum]
60 Day Email		[InvBalance]

VARIABLES

[CustName]
[InvNum]
[InvBalance]
[PenaltyAmt]
[DaysOverdue]
[TotalDue]

EMAIL MESSAGE

Hi [CustName],
I was just check
overdue balance
A penalty amou
the total owed
[DaysOverdue]
Your attention
Best Regards,
Fred Fredders
AR Manager

From: randy@ExcelForFreelancers.com
To: John_James@gmail.com
Subject: John James, 60 Day Overdue Notice For Invoice #6

Hi John James,
I was just checking our records and I see that you have an overdue balance on Invoice 6 for \$974.39
A penalty amount of \$50.00 has been assessed bringing the total owed to \$1,024.39 due to the invoice being 64 days overdue.
Your attention to this matter is greatly appreciated.
Best Regards,
Fred Fredders
AR Manager

EXCLUSIVE FOR
SILVER & GOLD
MEMBERS

FEATURE, FIX,
OR FOCUS



Excel For Freelancers

How To Create A Fully Automated Payment Reminder & Penalty System In Excel



[DOWNLOAD
APPLICATION](#)



[VIEW
TRAINING](#)

by: *Randy Austin*

ABOUT THE AUTHOR

A two-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With over 442,000 YouTube subscribers, 33,000,000 video views, 250+ comprehensive training videos, and a thriving 60,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

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



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



Microsoft®
Most Valuable
Professional



OUR COURSES & PRODUCTS



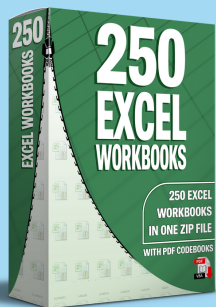
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



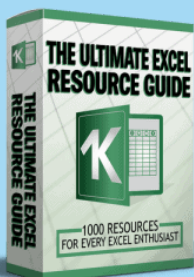
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



Incredible Package of 250 of my BEST Applications now with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "250 Workbook Library".

[Click here to learn more](#)



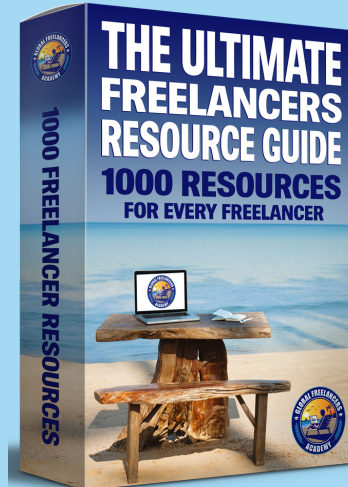
With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)



This incredible 13-hour freelancer masterclass will teach anyone how to be a successful freelancer with my proven 9-Phase 'Financial Freedom Roadmap' and includes 30+ downloads and exercises along with a ton of resources for anyone to reach their goals through freelancing.

[Click here to learn more](#)



I've put together all the Freelancing essentials, from freelance tools to freelance templates, in this Ultimate Freelancer's Resource Guide for freelancers at any stage in their career. With 1,000 Live Links, and a single click-to-update application, you will always have the most current and up-to-date information at your fingertips.

[Click here to learn more](#)

Table of Contents

Projects.....	3
VBAProject.....	3
Documents.....	3
Admin.....	3
(Declarations).....	3
Worksheet_Change [Sub].....	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
PayReminders.....	16
(Declarations).....	16
Worksheet_Change [Sub].....	16
Worksheet_SelectionChange [Sub].....	16
PenaltyLog.....	17
(Declarations).....	17
RemindersList.....	18
(Declarations).....	18
RemindLog.....	19
(Declarations).....	19
Sheet1.....	20
(Declarations).....	20
ThisWorkbook.....	21
(Declarations).....	21
Forms.....	22
AddCustForm.....	22
(Declarations).....	22
CancelBtn_Click [Sub].....	22
SaveBtn_Click [Sub].....	22
Modules.....	23
Admin_Menu_Macros.....	23
(Declarations).....	23
AppEvents_Start [Sub].....	23
AppEvents_Stop [Sub].....	23
Company_AddLogo [Sub].....	23
EditMode [Sub].....	23
Invoice_Macros.....	24
(Declarations).....	24
Customer_AddUpdate [Sub].....	24
Invoice_Delete [Sub].....	24
Invoice_Load [Sub].....	24
Invoice_New [Sub].....	25
Invoice_NewPage [Sub].....	25
Invoice_NextInvoice [Sub].....	25
Invoice_NextPage [Sub].....	26
Invoice_PageLoad [Sub].....	26
Invoice_PrevInvoice [Sub].....	26
Invoice_PrevPage [Sub].....	27
Invoice_SavedMsg [Sub].....	27

Table of Contents

Invoice_SaveUpdate [Sub]	27
PaymentReminders.....	29
(Declarations)	29
PayReminder_AddNew [Sub].....	29
PayReminder_CheckAndSend [Sub]	29
PayReminder_Delete [Sub].....	31
PayReminder_Load [Sub]	31
PayReminder_LoadList [Sub]	31
PayReminder_SaveUpdate [Sub]	32
PrintAttachToPDF [Sub].....	32
Payments_Macros	34
(Declarations)	34
Payments_AddNew [Sub]	34
Payments_Delete [Sub]	34
Payments_Load [Sub].....	34
Payments_LoadOpenInvoices [Sub]	35
Payments_Next [Sub]	35
Payments_Previous [Sub].....	36
Payments_SaveUpdate [Sub]	36
PopUp_Calendar.....	38
(Declarations)	38
CalCol [Sub]	38
CalendarHide [Sub].....	38
CalendarShow [Sub]	38
CalFormulaReplacement [Sub]	39
CheckForSheet [Sub].....	39
CreateCalSht [Sub].....	40
GroupCal [Sub]	42
MacroLinkRemover [Sub]	42
NextMonth [Sub]	43
NextYear [Sub]	43
PrevMonth [Sub]	43
PrevYear [Sub]	44
ReplaceCalendar [Sub].....	44
SelectDay [Sub]	44
ShowSettings [Sub]	44
UnGroupCal [Sub]	44
Print_Macros.....	46
(Declarations)	46
Invoice_Print [Sub].....	46
Invoice_SaveAsPDF [Sub].....	46

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("N19:N28" )) Is Nothing And Range("L" & Target.Row).
5         Value <> Empty Then
6         If Target.Value = "Percent of Bal." Then
7             Range("O" & Target.Row).NumberFormat = "0.0%"
8         Else
9             Range("O" & Target.Row).NumberFormat = "$#.00"
10        End If
11    End If
12 End Sub
13
14 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
15     If Target.CountLarge > 1 Then Exit Sub
16
17     'On Default Invoice Status Selection
18     If Not Intersect(Target, Range("F6:F11" )) Is Nothing And Range("E" & Target.Row).
19         Value <> Empty Then
20         Range("F6:F11" ).ClearContents
21         Target.Value = Chr(252)
22         Range("E6" ).Select
23     End If
24
25     'On Default Payment Terms Selection
26     If Not Intersect(Target, Range("J6:J11" )) Is Nothing And Range("H" & Target.Row).
27         Value <> Empty Then
28         Range("J6:J11" ).ClearContents
29         Target.Value = Chr(252)
30         Range("H6" ).Select
31     End If
32 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        End If
53
54        'On Any Invoice Item Detail Change, Save To Database (not on Invoice Load)
55        If Not Intersect(Target, Range("J10:P44" )) Is Nothing And Range("B1" ).Value =
56            False Then
57            SelRow = Target.Row 'Set Selected Row
58            'Update Invoice Items DB
59            If Range("Q" & SelRow).Value = Empty Then 'New Invoice Item

```

1 2 3

```

1 2 3
52   InvItemRow = InvoiceItems.Range("A9999" ).End(xlUp).Row + 1 'First Avail.
    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 & "_Lb1" )
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

```

```

1
101 If Shapes("Calendar").Visible = True Then CalendarHide 'Run macro to hide calendar
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 +
Columns("N").ColumnWidth + Columns("O").ColumnWidth + Columns("P").
ColumnWidth

```

1 2 3 4

```

1 2 3 4
152 DescWidth = Columns("L").ColumnWidth
153 If InvWidth > 546.5 Then Columns("L").ColumnWidth = (DescWidth - (InvWidth
- 93.6))
154 If InvWidth < 546.5 Then Columns("L").ColumnWidth = (DescWidth + (93.6 -
InvWidth))
155 End If
156
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
exist
166 If ItemNm <> "Logo" Then
167 Shapes("SampleInvShp").Duplicate.Name = ItemNm 'Create Shape If
missing
168 Else 'Logo
169 If Admin.Range("C10").Value = Empty Or Dir(Admin.Range("C10").Value,
vbDirectory) = "" Then GoTo CheckForLbl
'On Error Resume Next
170 Pictures.Insert(Admin.Range("C10").Value).Name = ItemNm 'Set Logo
Picture
171 'On Error GoTo 0
172 End If
173 Set ItemShp = Shapes(ItemNm)
174 With ItemShp
175 If ItemNm <> "Logo" Then .DrawingObject.Formula = "=" & Cells(ItemRow,
ItemCol + 1).Address
176 .Left = Range("L1").Left
177 .Top = Range("L5").Top
178 .Placement = xlMove 'Move but not size with cells (columns)
179 End With
180 End If
181 CheckForLbl:
182 'Check For Item Label Shape
183 On Error Resume Next
184 Set ItemLblShp = Shapes(ItemNm & "_Lbl" )
185 On Error GoTo 0
186 If ItemLblShp Is Nothing Then 'Create & Position Value Shape if it does not
exist
187 Shapes("SampleInvShp").Duplicate.Name = ItemNm & "_Lbl" 'Create Shape If
missing
188 Set ItemLblShp = Shapes(ItemNm & "_Lbl" )
189 With ItemLblShp
190 .DrawingObject.Formula = "=" & Cells(ItemRow, ItemCol).Address
191 .Left = Range("L1").Left - 50
192 .Top = Range("L5").Top
193 .Placement = xlMove
194 End With
195 End If
196
197
198 'Group Shapes if group does not exist
199 On Error Resume Next
200 Set GrpShp = Shapes(ItemNm & "_Grp" )
201 On Error GoTo 0
202 If GrpShp Is Nothing Then 'Group Shapes

```

```
1 2 3 4
203 Shapes.Range(Array(ItemShp.Name, ItemLblShp.Name)).Group.Name = ItemNm &
    "_Grp"
204 Shapes(ItemNm & "_Grp" ).Placement = xlMove 'Set to Move but not size with
    cells
205 End If
206
207 If Target.Value = Empty Or Target.Offset(0, 1).Value = Empty Then 'Ungroup
    and Hide on Empty Label
208     On Error Resume Next
209     Shapes(ItemNm & "_Grp" ).Ungroup
210     On Error Resume Next
211     ItemLblShp.Visible = msoFalse
212 Else
213     ItemLblShp.Visible = msoCTrue 'Show/Hide Invoice Label
214 End If
215 If Target.Value = Empty Then 'Hidden
216     ItemShp.Visible = msoFalse
217 Else 'Visible
218     ItemShp.Visible = msoCTrue 'Show/Hide Invoice Field Value
219 End If
220 End If
221 Range("F2" ).Select
222 End If
223 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 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("G8" )) Is Nothing And Range("G8" ).Value <> Empty
5         Then
6             If Range("B6" ).Value <> Empty Then
7                 Dim PenRow As Long
8                 Dim PenType As String
9                 PenRow = Range("B6" ).Value 'Penalty Row
10                If Admin.Range("N" & PenRow).Value = "Percent of Bal." Then PenType =
11                    "percent" Else PenType = "dollars"
12                Range("I8" ).Value = Admin.Range("M" & PenRow).Value & " days overdue at"
13                Range("J8" ).Value = Admin.Range("O" & PenRow).Value 'percent or amount
14                Range("K8" ).Value = PenType 'Set Penalty Type (percent/amount)
15                If PenType = "percent" Then Range("J8" ).NumberFormat = "0.0%" Else Range(
16                    "J8" ).NumberFormat = "$#.00"
17            End If
18        End If
19    End Sub
20
21 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
22     If Target.CountLarge > 1 Then Exit Sub
23     If Not Intersect(Target, Range("D5:D99" )) Is Nothing And Range("D" & Target.Row).
24         Value <> Empty Then
25         Range("B2" ).Value = Target.Row 'Trigger CF
26         Range("B3" ).Value = Range("C" & Target.Row).Value 'Reminder ID
27         PayReminder_Load 'Run Macro to load reminder
28     End If
29 End Sub
```

1	Option Explicit
2	

1	Option Explicit
2	

1	Option Explicit
2	

1 Option Explicit

2

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()
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_NewPage()
99     Invoice_SaveUpdate 'Save New Last page limit
100    Invoice.Range("B11" ).Value = Invoice.Range("B12" ).Value + 1 'Set Selected page to
101    new page
102    Invoice.Range("B12" ).Value = Invoice.Range("B12" ).Value + 1 ' Increment page to +
103    1
104    Invoice_PageLoad
105    Invoice_SaveUpdate 'Save New Last page limit
106 End Sub

107 Sub Invoice_NextInvoice()
108     With Invoice
109         Dim MaxInvNumb As Long
110         On Error Resume Next

```

```

1 2
107 MaxInvNumb = Application.WorksheetFunction.Max(InvoiceList.Range("Invoice_ID" ))
108 On Error GoTo 0
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("A9999" ).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

```

```

1 2
163 On Error Resume Next
164 MinInvNumb = Application.WorksheetFunction.Min(InvoiceList.Range("Invoice_ID" ))
165 On Error GoTo 0
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("A9999" ).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("A9999" ).End(xlUp).Row + 1 'First Avail Row

```

1 2 3

```
1 2 3
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)"
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, ResultRow As Long, RemRow As Long, RemCol
  As Long
3
4 Sub PayReminder_AddNew()
5     PayReminders.Range("B2,B3,G4:H4,K4,G6,K6,G8:H8,K8,G10:K10,G12:K33").ClearContents
  'Clear Out Form Fields
6     PayReminders.Range("K4").Value = "Active" 'Set Default
7 End Sub
8
9
10 Sub PayReminder_CheckAndSend()
11     Dim DaysOverdue As Long, InvNumb As Long, DueDays As Long, LastInvRow As Long,
  LastInvResultRow As Long, ODInvs As Long
12     Dim InvResultRow As Long, CustRow As Long, LogRow As Long, PenaltyRow As Long,
  RemCount As Long
13     Dim BalThresh As Double, CustBal As Double, InvBal As String, Penalty As Double,
  PenaltyAmt As Double, TotalDue As String
14     Dim EmSubj As String, EmMesg As String, Subj As String, Mesg As String, Reminder As
  String
15     Dim CustNm As String, PenaltyType As String, CustEmail As String, PenaltyNm As
  String, RemindID As String
16     Dim OutMail As Object, OutApp As Object
17
18     With RemindersList
19         PayReminders.Range("B8").ClearContents
20         RemCount = 0 'Set reminder to 0 initially
21         LastRow = .Range("A99999").End(xlUp).Row
22         If LastRow < 4 Then Exit Sub
23         .Range("T3:AC999").ClearContents 'Clear any prior results
24         .Range("A3:H" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
  "Q2:Q3"), CopyToRange:=.Range("T2:AA2"), Unique:=True
25         LastResultRow = .Range("T99999").End(xlUp).Row
26         If LastResultRow < 3 Then Exit Sub
27         .Range("AB3:AC" & LastResultRow).Formula = .Range("AB1:AC1").Formula 'Bring
  down Penalty Details (from Formula)
28         LastInvRow = InvoiceList.Range("A9999").End(xlUp).Row 'Last Invoice Row
29         If LastInvRow < 3 Then Exit Sub 'Exit on no invoices
30         Set OutApp = CreateObject("Outlook.application") 'Create Outlook application
31         For ResultRow = 3 To LastResultRow 'Loop Through Active Payment Reminders
32             InvoiceList.Range("AA3:AK9999").ClearContents 'Clear Invoice result
33             RemindID = .Range("T" & ResultRow).Value 'Reminder ID
34             Reminder = .Range("Y" & ResultRow).Value 'Reminder Name
35             PenaltyNm = .Range("Y" & ResultRow).Value 'Penalty Name
36             DueDays = .Range("W" & ResultRow).Value 'Due Days
37             BalThresh = .Range("X" & ResultRow).Value 'Balance threshold
38             Subj = .Range("Z" & ResultRow).Value 'Email Subject
39             Mesg = .Range("AA" & ResultRow).Value 'Email Message
40             PenaltyType = .Range("AB" & ResultRow).Value 'Penalty Type
41             Penalty = .Range("AC" & ResultRow).Value 'Perc. or $ Amount
42             InvoiceList.Range("W3").Value = "<=" & Int(CDbl(Date)) - DueDays 'Create
  Due Date Criteria
43             InvoiceList.Range("X3").Value = ">" & BalThresh 'Balance Criteria
44             InvoiceList.Range("Y2").Value = RemindID 'Used in Sent/Not Sent Formula
45             With InvoiceList
46                 .Range("AA3:AM9999").ClearContents 'Clear Prior Results
47                 .Range("A2:J" & LastInvRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.
  Range("W2:X3"), CopyToRange:=.Range("AA2:AJ2"), Unique:=True
48                 LastInvResultRow = .Range("AA99999").End(xlUp).Row
49                 If LastInvResultRow < 3 Then GoTo NextResult

```

1 2 3 4

```

1 2 3 4
50 .Range("AK3:AM" & LastInvResultRow).Formula = .Range("AK1:AM1").Formula
    'Sent/Not Sent Formula
51 For InvResultRow = 3 To LastInvResultRow 'loop through invoice result
52     If .Range("AK" & InvResultRow).Value = "Not Sent" Then
53         InvBal = Format(.Range("AJ" & InvResultRow).Value, "$#,##0.00")
            'Balance Amount
54         InvNum = .Range("AA" & InvResultRow).Value 'Invoice Number
55         CustNm = .Range("AC" & InvResultRow).Value 'Customer Name
56         CustBal = Format(.Range("AL" & InvResultRow).Value, "$#,##0.00")
            'Customer Balance
57         ODInvs = .Range("AM" & InvResultRow).Value 'Overdue Invoices
58         On Error Resume Next
59         CustRow = Customers.Range("Cust_Name").Find(CustNm, , xlValues,
            xlWhole).Row
60         On Error GoTo 0
61         If CustRow <> 0 Then CustEmail = Customers.Range("H" & CustRow).
            Value Else CustEmail = "" 'Set Customer Email
62         If PenaltyType = "Percent of Bal." Then PenaltyAmt = Penalty *
            InvBal Else PenaltyAmt = Penalty
63         PenaltyAmt = Format(PenaltyAmt, "$#,##0.00") 'Format Penalty Amount
64         DaysOverdue = Date - .Range("AF" & InvResultRow).Value 'Determine
            Days overdue
65         TotalDue = Format(PenaltyAmt + InvBal, "$#,##0.00") 'Total Due
            (formatted)
66         EmSubj = Replace(Replace(Replace(Replace(Replace(Replace(Subj,
            "[CustName]", CustNm), "[InvNum]", InvNum), "[InvBalance]",
            InvBal), "[PenaltyAmt]", PenaltyAmt), "[DaysOverdue]", DaysOverdue)
            , "[TotalDue]", TotalDue)
67         EmMesg = Replace(Replace(Replace(Replace(Replace(Replace(Replace(
            Replace(Mesg, "[CustName]", CustNm), "[InvNum]", InvNum),
            "[InvBalance]", InvBal), "[PenaltyAmt]", PenaltyAmt),
            "[DaysOverdue]", DaysOverdue), "[TotalDue]", TotalDue),
            "[CustBalance]", Format(CustBal, "$#,##0.00")), "[#ODInvoices]",
            ODInvs)
68
69         'Check For Invoice Attachment
70         If PayReminders.Range("G11").Value = "Yes" And InvNum <> 0 Then
71             Invoice.Range("F2").Value = InvNum 'Add Invoice in Search to
                Load Invoice
72             PrintAttachToPDF
73         End If
74         Set OutMail = OutApp.CreateItem(0)
75         With OutMail
76             .To = CustEmail 'To email
77             .Subject = EmSubj
78             .Body = EmMesg
79             If PayReminders.Range("B8").Value <> Empty Then .Attachments.Add
                PayReminders.Range("B8").Value 'Attach PDF File Path
80             .Display 'Use .Send to send without displaying
81         End With
82
83         'Add To Reminder Log
84         LogRow = RemindLog.Range("A9999").End(xlUp).Row + 1 'First Avail
            row
85         RemindLog.Range("A" & LogRow).Value = Date 'Current Date
86         RemindLog.Range("B" & LogRow).Value = CustNm 'Customer Name
87         RemindLog.Range("C" & LogRow).Value = RemindID 'Reminder ID
88         RemindLog.Range("D" & LogRow).Value = Reminder 'Reminder
89         RemindLog.Range("E" & LogRow).Value = PenaltyNm 'Penalty Name

```

```

1 2 3 4 5 6
90 RemindLog.Range("F" & LogRow).Value = InvNumb 'Invoice Number
91 RemindLog.Range("G" & LogRow).Value = EmSubj 'Email Subject
92 RemindLog.Range("H" & LogRow).Value = EmMesg 'Email Message
93 RemindLog.Range(LogRow & ":" & LogRow).WrapText = False
94
95     If PenaltyAmt <> 0 Then 'Add To Penalty Log
96         PenaltyRow = PenaltyLog.Range("A9999" ).End(xlUp).Row + 1
97         'Penalty Row
98         PenaltyLog.Range("A" & PenaltyRow).Value = Date 'Set Current
99         Date
100        PenaltyLog.Range("B" & PenaltyRow).Value = InvNumb 'Set Invoice
101        Number
102        PenaltyLog.Range("C" & PenaltyRow).Value = CustNm 'Set Customer
103        Name
104        PenaltyLog.Range("D" & PenaltyRow).Value = PenaltyNm 'Penalty
105        Name
106        PenaltyLog.Range("E" & PenaltyRow).Value = PenaltyAmt 'Penalty
107        Amount
108    End If
109    RemCount = RemCount + 1 'Increment Reminder Counter
110 End If
111 NextResult:
112 Next InvResultRow
113 End With
114 Next ResultRow
115 PayReminders.Activate
116 MsgBox RemCount & " reminder(s) Sent"
117 End With
118 End Sub
119
120 Sub PayReminder_Delete()
121     If MsgBox("Are you sure you want to delete this Payment Reminder?" , vbYesNo,
122         "Delete Payemnt Reminder" ) = vbNo Then Exit Sub
123     With PayReminders
124         If .Range("B4" ).Value = Empty Then GoTo NotSaved
125         RemRow = .Range("B4" ).Value 'Set Row
126         RemindersList.Range(RemRow & ":" & RemRow).EntireRow.Delete 'Delete DB Row
127         NotSaved:
128         PayReminder_AddNew
129         PayReminder_LoadList
130     End With
131 End Sub
132
133 Sub PayReminder_Load()
134     With PayReminders
135         .Range("G4:H4,K4,G6,K6,G8:H8,K8,G10:K10,G12:K33" ).ClearContents 'Clear Out Form
136         Fields
137         If .Range("B4" ).Value = Empty Then
138             MsgBox "Please select a correct reminder"
139             Exit Sub
140         End If
141         RemRow = .Range("B4" ).Value
142         For RemCol = 2 To 8
143             .Range(RemindersList.Cells(1, RemCol).Value).Value = RemindersList.Cells(
144                 RemRow, RemCol).Value 'Add data to form field from db
145         Next RemCol
146     End With
147 End Sub
148
149 Sub PayReminder_LoadList()
150     1

```

```

1
141 PayReminders.Range("C5:D9999").ClearContents 'Clear out any previous reminders &
    ID's
142 With RemindersList
143     LastRow = .Range("A99999").End(xlUp).Row
144     If LastRow < 4 Then Exit Sub
145     On Error Resume Next
146     .Names("Criteria").Delete
147     On Error GoTo 0
148     .Range("A3:B" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
        "M2:N2"), Unique:=True
149     LastResultRow = .Range("M99999").End(xlUp).Row
150     If LastResultRow < 3 Then Exit Sub
151     If LastResultRow < 4 Then GoTo NoSort
152     With .Sort
153         .SortFields.Clear
154         .SortFields.Add Key:=RemindersList.Range("N3"), SortOn:=xlSortOnValues, Order
            :=xlAscending, DataOption:=xlSortNormal 'Sort
155         .SetRange RemindersList.Range("M3:N" & LastResultRow) 'Set Range
156         .Apply 'Apply Sort
157     End With
158     NoSort:
159     PayReminders.Range("C5:D" & LastResultRow + 2).Value = .Range("M3:N" &
        LastResultRow).Value 'Bring Over Unique Reminders
160
161 End With
162 End Sub
163 Sub PayReminder_SaveUpdate()
164     With PayReminders
165         If .Range("G4").Value = Empty Then
166             MsgBox "Please make sure to add a reminder name"
167             Exit Sub
168         End If
169
170         If .Range("B4").Value = Empty Then 'New Payment Reminder
171             RemRow = RemindersList.Range("A9999").End(xlUp).Row + 1 ' First Avail Row
172             .Range("B3").Value = .Range("B5").Value 'Next Rem. ID
173             RemindersList.Range("A" & RemRow).Value = .Range("B3").Value 'Nexst Rem. ID
174
175         Else 'Existing
176             RemRow = .Range("B4").Value 'Existing reminder Row
177         End If
178         For RemCol = 2 To 8
179             RemindersList.Cells(RemRow, RemCol).Value = .Range(RemindersList.Cells(1,
                RemCol).Value).Value 'Add Data to DB
180         Next RemCol
181         RemindersList.Range(RemRow & ":" & RemRow).WrapText = False
182         PayReminder_LoadList
183     End With
184 End Sub
185
186 Sub PrintAttachToPDF()
187     Dim Pages As String, FilePath As String, SheetsArr() As String
188     Dim LastPage As Long, PageNum As Long
189     Application.ScreenUpdating = False
190     PayReminders.Range("B8").ClearContents
191     With Invoice
192         LastPage = .Range("B12").Value 'Set Last Page
193         FilePath = ThisWorkbook.Path & "\Invoice_" & Invoice.Range("B4").Value &
            ".pdf"

```

1 2

```
1 2
194 For PageNumb = 1 To LastPage
195     .Range("B11" ).Value = PageNumb
196     Invoice_PageLoad 'Load Page
197     .Select
198     .Copy After:=Sheets(Sheets.Count)
199     ActiveSheet.Name = "Page" & PageNumb
200     If PageNumb <> LastPage Then
201         Pages = Pages & "Page" & PageNumb & "," 'Build Sheet name string
202     Else
203         Pages = Pages & "Page" & PageNumb 'no comma on last page
204     End If
205 Next PageNumb
206 SheetsArr = Split(Pages, ",") 'Create Sheet array
207 Sheets(SheetsArr).Select
208 ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, Filename:=FilePath, _
209     Quality:=xlQualityStandard, IncludeDocProperties:=True,
    IgnorePrintAreas _
210     :=False, OpenAfterPublish:=False
211 PayReminders.Range("B8" ).Value = FilePath 'Set Attach File Path
212 'Delete Temporary Sheets
213 Sheets(SheetsArr).Select
214 Application.DisplayAlerts = False
215 Sheets(SheetsArr).Delete
216 Application.DisplayAlerts = True
217 End With
218 Application.ScreenUpdating = True
219
220 End Sub
221
222
```

```

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

```

1 2 3

```

54     1 2 3
55     Next PayCol
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    End With

```

```

107
108
109     If MaxPayID = PayID Then 'Last Payment
110         MsgBox "You are on the last payment"
111         Exit Sub
112     End If
113
114     .Range("B3" ).Value = PaymentList.Range("A" & PayRow).Value 'Set Payment ID
115     Payments_Load 'Load Payment
116 End With
117 End Sub
118
119 Sub Payments_Previous()
120     Dim MinPayID As Long, PayID As Long
121     With Payments
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

```



```
165     PayRow = .Range("B4" ).Value
166     End If
167     For PayCol = 2 To 6
168         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
           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$4,Cal
177 Months,0),1)," & Chr(34) & Chr(34) & ")"
178 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
    ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A
179 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
180 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
    ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A
181 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
182 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
    ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A
183 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
184 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
    ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A
185 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
186 .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
    ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A
187 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
188 .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
    ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A
189 $4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
190 .Range("B2" ).Value = "=H1+1"
191 .Range("C2" ).Value = "=B2+1"
192 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
193 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
194 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
195 & Chr(34) & Chr(34) & ",H4+1)"
196 .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) &
197 ",B5+1)," & Chr(34) & Chr(34) & ")"
198 .Range("B6" ).Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) &
199 ",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) &
200 ")")
201 .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) &
202 ",I5+1)," & Chr(34) & Chr(34) & ")"
203 .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
204
205 'Set format to Single Day
206 .Range("B1:H6" ).NumberFormat = "d"
207
208 'Add in relative Day #'s
209 .Range("B7" ).Value = "1"
210 .Range("C7" ).Value = "2"
211 .Range("B8" ).Value = "8"
212 .Range("C8" ).Value = "9"
213 .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
214 .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
215 GroupCal
216 End With
217 Exit Sub
218 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
```

—, 7, 15, 33, 42, 46

A

Activate, 31, 40, 46
 ActiveCell, 38, 39, 44
 ActiveSheet, 33, 38-40, 42-46
 ActiveWorkbook, 39-41
 ActSht, 40
 Add, 23, 24, 30, 32, 34, 40, 41
 AddCustForm, 22, 24
 Address, 8, 39, 40
 Admin, 6, 8, 16, 23, 25
 AdvancedFilter, 24, 26, 29, 32, 34, 35
 After, 33, 40, 46
 AllowMultiSelect, 23
 AppEvents_Start, 23-25, 28
 AppEvents_Stop, 23-25, 27
 Application, 5, 22, 23, 26-28, 32, 33, 35, 36, 38, 44, 46
 Apply, 24, 32, 34
 Array, 9, 38, 39, 42-44
 Attachments, 30
 AutoFill, 41

B

BalThresh, 29
 Body, 30
 Bold, 38, 39, 43
 BtnMov, 23

C

Cal, 6
 CalCol, 38, 40
 Calculation, 23
 CalendarHide, 7, 38
 CalendarShow, 7, 38
 CalFormulaReplacement, 39
 Caller, 28, 38, 44
 CancelBtn_Click, 22
 Cells, 6-8, 22, 24, 25, 28, 31, 32, 34, 37, 39, 40, 44
 CheckForLbl, 8
 CheckForSheet, 7, 39
 Chr, 3, 7, 15, 25, 37, 40, 41
 Clear, 24, 32, 34
 ClearContents, 3, 5, 7, 15, 25, 26, 29, 31, 32, 34, 35
 ColCnt, 40
 ColNum, 39
 ColNumb, 44
 Column, 6
 Columns, 7, 8
 ColumnWidth, 7, 8
 Company_AddLogo, 23
 Control, 22, 24
 Controls, 22, 24
 Copy, 33, 40, 44, 46
 CopyToRange, 24, 26, 29, 32, 34, 35
 Count, 33, 40, 46
 CountLarge, 3, 5, 6, 15, 16
 CreateCalSht, 40
 CreateItem, 30
 CreateObject, 29
 CreateWS, 39, 40
 CriteriaRange, 24, 26, 29, 34, 35
 CustBal, 29, 30
 CustCol, 22, 24
 CustEmail, 29, 30

CustNm, 29-31
 Customer_AddUpdate, 24
 CustomerFld, 22, 24
 Customers, 22, 24, 30
 CustRow, 22, 24, 29, 30

D

DataOption, 24, 32, 34
 Day, 39
 DayCnt, 40
 DayName, 38, 39, 43
 DayNum, 38, 39
 DayNumb, 44
 DaysOverdue, 29, 30
 Delay, 27
 Delete, 24, 31-34, 46
 DescWidth, 7, 8
 Destination, 41
 Dir, 8
 Display, 30
 DisplayAlerts, 33, 46
 DoEvents, 27
 DrawingObject, 8, 39, 40
 DueDays, 29
 Duplicate, 8

E

EditMode, 23
 EmMsg, 29-31
 Empty, 3, 5-9, 15, 16, 22, 24-27, 30-32, 34-38, 43, 44
 EmSubj, 29-31
 EnableEvents, 5, 23
 EntireColumn, 7
 EntireRow, 24, 31, 34
 Explicit, 3-5, 10-24, 29, 34, 46
 ExportAsFixedFormat, 33, 46

F

Field1, 22
 FieldRow, 24, 25, 28
 FileDialog, 23
 Filename, 33, 46
 FilePath, 32, 33, 46
 Fill, 23, 27, 38, 39, 43
 Filters, 23
 Find, 5, 6, 25-27, 30
 Font, 38, 39, 43
 ForeColor, 23, 38, 39, 43
 Format, 30
 Formula, 8, 29, 30, 35, 39, 40

G

Group, 9, 42
 GroupCal, 41, 42
 GrpShp, 6, 8

H

HeadShp, 24
 Hidden, 7
 Hide, 22

I

i, 27
 IgnorePrintAreas, 33, 46
 IncludeDocProperties, 33, 46
 Insert, 8
 InStr, 39, 43
 Int, 29
 Intersect, 3, 5-7, 15, 16

InvBal, 29, 30
 InvItemRow, 5, 6, 24, 26
 InvNumb, 24, 26, 27, 29-31
 Invoice, 22-27, 30, 32, 46
 Invoice_Delete, 24
 Invoice_Load, 5, 25-27
 Invoice_New, 24, 25
 Invoice_NewPage, 25
 Invoice_NextInvoice, 25
 Invoice_NextPage, 26
 Invoice_PageLoad, 25-27, 33, 46
 Invoice_PrevInvoice, 26
 Invoice_PrevPage, 27
 Invoice_Print, 46
 Invoice_SaveAsPDF, 46
 Invoice_SavedMsg, 27, 28
 Invoice_SaveUpdate, 25, 27, 46
 InvoiceItems, 6, 24, 26
 InvoiceList, 24-29, 35
 InvResultRow, 29-31
 InvRow, 24-28
 InvWidth, 7, 8
 IsDate, 38
 ItemCol, 5-8
 ItemDBRow, 5, 24
 ItemLblShp, 6-9
 ItemNm, 6, 8, 9
 ItemRow, 5, 7, 8, 24
 Items, 5
 ItemShp, 6-9
 ItemTypeRow, 5-8

K

Key, 24, 32, 34

L

LastInvResultRow, 29, 30
 LastInvRow, 29
 LastItemRow, 24
 LastPage, 32, 33, 46
 LastPayItemRow, 34, 37
 LastResultRow, 24, 26, 29, 32, 34, 35
 LastRow, 24, 26, 29, 32, 34, 35
 Left, 8, 23, 39, 43
 Len, 43
 LogoPic, 23
 LogRow, 29-31

M

MacroLink, 42, 43
 MacroLinkRemover, 39, 42
 Max, 22, 26, 35
 MaxInvNumb, 25, 26
 MaxPayID, 35, 36
 Msg, 29, 30
 Min, 27, 36
 MinInvNumb, 26, 27
 MinPayID, 36
 Month, 38, 40
 MovInc, 23
 MsgBox, 5, 22, 24-27, 31, 32, 34-36, 39, 42
 msoCTrue, 6, 9, 27, 39, 42
 msoFalse, 6, 9, 27, 38, 43, 44
 msoFileDialogFilePicker, 23
 msoTrue, 39

N

Name, 8, 9, 33, 40-42, 46
 Names, 32, 41
 NewLink, 42, 43

NextMonth, 43
 NextResult, 29, 31
 NextShp, 43
 NextYear, 43
 NoCal, 38, 39, 41
 NoData, 35
 NoItems, 26
 NoSelection, 23
 NoSort, 32
 NotSaved, 24, 31, 34
 NumberFormat, 3, 16, 41

O

ODInvs, 29, 30
 Offset, 9, 39, 44
 OnAction, 40, 43
 OpenAfterPublish, 33, 46
 Order, 24, 32, 34
 OutApp, 29, 30
 OutMail, 29, 30

P

PageNum, 32, 33, 46
 Pages, 32, 33, 46
 Paste, 40, 44
 Path, 32, 46
 PayCol, 34, 35, 37
 PayID, 35, 36
 PayItemDBRow, 34, 37
 PayItemRow, 34, 37
 PayItems, 34, 35, 37
 PaymentList, 34-37
 Payments, 34-36
 Payments_AddNew, 34
 Payments_Delete, 34
 Payments_Load, 34, 36
 Payments_LoadOpenInvoices, 15, 35
 Payments_Next, 35
 Payments_Previous, 36
 Payments_SaveUpdate, 36
 PayReminder_AddNew, 29, 31
 PayReminder_CheckAndSend, 29
 PayReminder_Delete, 31
 PayReminder_Load, 16, 31
 PayReminder_LoadList, 31, 32
 PayReminder_SaveUpdate, 32
 PayReminders, 29-33
 PayRow, 34-37
 Penalty, 29, 30
 PenaltyAmt, 29-31
 PenaltyLog, 31
 PenaltyNm, 29-31
 PenaltyRow, 29, 31
 PenaltyType, 29, 30
 PenRow, 16
 PenType, 16
 PicPath, 23
 Pictures, 8
 Placement, 8, 9, 39, 42
 PrevMonth, 43
 PrevYear, 44
 PrintAttachToPDF, 30, 32
 PrintOut, 46

Q

Quality, 33, 46

R

Range, 3, 5-9, 15, 16, 22-44, 46
 RefersTo, 41

RemCol, 29, 31, 32
 RemCount, 29, 31
 Reminder, 29, 30
 RemindersList, 29, 31, 32
 RemindID, 29, 30
 RemindLog, 30, 31
 RemRow, 29, 31, 32
 Replace, 30, 44
 ReplaceCalendar, 6, 44
 ResultRow, 24, 26, 29, 31, 34
 RGB, 23, 38, 39, 43
 Right, 43
 RoundUp, 44
 Row, 3, 5-8, 15, 16, 22, 24-27, 29-32, 34-37
 RowCnt, 40
 RowNum, 39
 RowNumb, 44

S

SaveBtn_Click, 22
 ScreenUpdating, 23, 32, 33, 46
 SelCell, 38, 39
 SelectDay, 44
 SelectedItems, 23
 Selection, 38, 42
 SelRow, 5, 6
 SetRange, 24, 32, 34
 Shape, 6, 24, 39, 42
 ShapeRange, 38, 42
 Shapes, 6-9, 23, 27, 38-40, 42-45
 Sheets, 33, 38-40, 43, 44, 46
 SheetsArr, 32, 33, 46
 Show, 23, 24
 ShowSettings, 44
 Shp, 39, 42, 43
 SingleRow, 24
 SkipSort, 34
 Sort, 24, 32, 34
 SortFields, 24, 32, 34
 SortOn, 24, 32, 34
 Split, 33, 43, 46
 SplitLink, 42, 43
 StartTime, 27
 StatusRow, 24, 25
 Subj, 29, 30
 Subject, 30

T

Target, 3, 5-9, 15, 16
 TermRow, 5, 6, 24, 25
 Text, 23
 TextFrame2, 23, 38, 39, 43
 TextRange, 23, 38, 39, 43
 ThisWorkbook, 32, 40, 43, 44, 46
 Timer, 27
 Title, 23
 Top, 8, 39
 TotalDue, 29, 30
 Transparency, 27

U

Ungroup, 9, 45
 UnGroupCal, 40, 44
 Unique, 24, 26, 29, 32, 34, 35
 Unload, 24

V

Value, 3, 5-9, 15, 16, 22-41, 43, 44, 46
 vbDirectory, 8
 vbNo, 24, 31, 34

vbYesNo, 24, 31, 34
 Visible, 6, 7, 9, 27, 38-40, 42-44

W

Worksheet, 39, 40
 Worksheet_Change, 3, 5, 15, 16
 Worksheet_SelectionChange, 3, 6, 15, 16
 WorksheetFunction, 22, 26, 27, 35, 36, 44
 WrapText, 31, 32
 ws, 39, 40

X

xlAscending, 32
 xlCalculationAutomatic, 23
 xlCalculationManual, 23
 xlDescending, 24, 34
 xlFillDefault, 41
 xlFilterCopy, 24, 26, 29, 32, 34, 35
 xlMove, 8, 9, 39, 42
 xlQualityStandard, 33, 46
 xlSheetHidden, 40
 xlSortNormal, 24, 32, 34
 xlSortOnValues, 24, 32, 34
 xlTypePDF, 33, 46
 xlUp, 6, 22, 24, 26, 27, 29-32, 34-37
 xlValues, 5, 6, 25-27, 30
 xlWhole, 5, 6, 25-27, 30

Y

Year, 38, 40

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