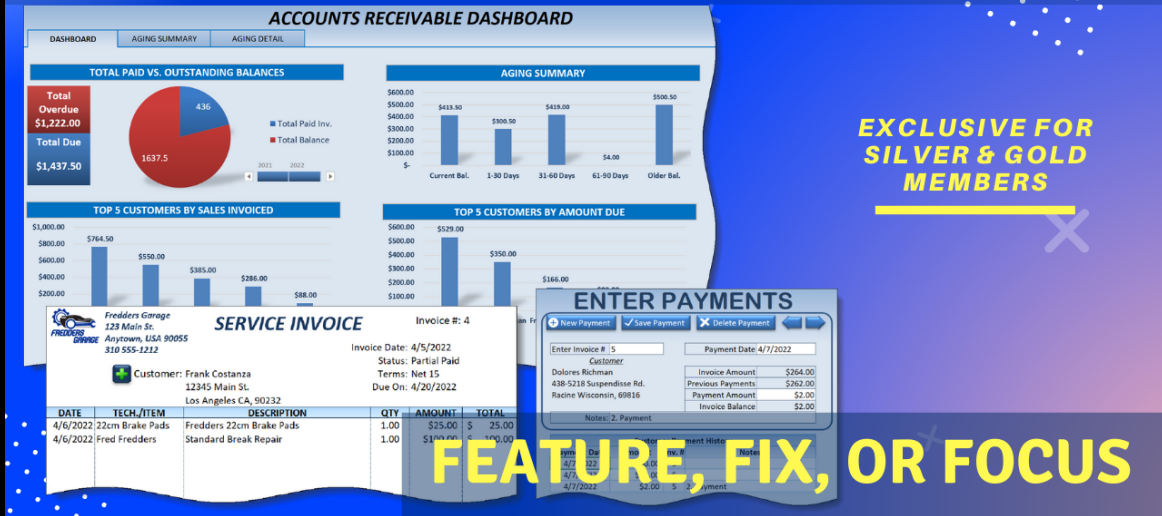


VBA SOURCE CODE BOOK

ACCOUNTS RECEIVABLE MANAGER- **UPDATED**



Excel For Freelancers

How To Create An Accounts Receivable Application And Manage Payments 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 300,000 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®
Most Valuable
Professional



OUR COURSES & PRODUCTS



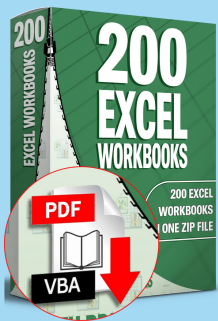
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[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 200 of my BEST Applications now with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "200 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](#)

Projects.....	3
VBAProject.....	3
Documents.....	3
Admin.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub].....	3
Customers.....	4
(Declarations).....	4
Dashboard.....	5
(Declarations).....	5
Worksheet_SelectionChange [Sub].....	5
Invoice.....	6
(Declarations).....	6
Worksheet_Change [Sub].....	6
Worksheet_SelectionChange [Sub].....	7
InvoiceItems.....	9
(Declarations).....	9
InvoiceList.....	10
(Declarations).....	10
Items.....	11
(Declarations).....	11
Payments.....	12
(Declarations).....	12
Worksheet_Change [Sub].....	12
Worksheet_SelectionChange [Sub].....	12
PmntsDB.....	13
(Declarations).....	13
RecurDB.....	14
(Declarations).....	14
Sheet1.....	15
(Declarations).....	15
Sheet6.....	16
(Declarations).....	16
Techs.....	17
(Declarations).....	17
ThisWorkbook.....	18
(Declarations).....	18
Forms.....	19
AddCustForm.....	19
(Declarations).....	19
CancelBtn_Click [Sub].....	19
SaveBtn_Click [Sub].....	19
Recurring.....	20
(Declarations).....	20
CancelBtn_Click [Sub].....	20
ClientNmLbl_Click [Sub].....	20
DeleteBtn_Click [Sub].....	20
RecField5_Change [Sub].....	20
RecField6_AfterUpdate [Sub].....	20
RecField6_Change [Sub].....	20
SaveBtn_Click [Sub].....	20
Modules.....	21
Dashboard_Macros.....	21
(Declarations).....	21
Aging_GoToInvoice [Sub].....	21
Aging_Refresh [Sub].....	21
Aging_ShowCustDetail [Sub].....	21
AgingDetail_Refresh [Sub].....	21
Dashboard_Refresh [Sub].....	22
Dashboard_SelectAgingDetails [Sub].....	22
Dashboard_TabChange [Sub].....	22
Invoice_Macros.....	24
(Declarations).....	24
Customer_AddNew [Sub].....	24
Invoice_Delete [Sub].....	24
Invoice_Load [Sub].....	24

Invoice_New [Sub]	25
Invoice_Print [Sub]	25
Invoice_SaveAsPDF [Sub]	25
Invoice_SavedMsg [Sub]	25
Invoice_SaveUpdate [Sub]	26
Pmnt_Macros	27
(Declarations)	27
Payment_CustomerPmntsRefresh [Sub]	27
Payment_Delete [Sub]	27
Payment_Load [Sub]	27
Payment_New [Sub]	27
Payment_Next [Sub]	28
Payment_Prev [Sub]	28
Payment_SavedMsg [Sub]	28
Payment_SaveUpdate [Sub]	29
PopUp_Calendar	30
(Declarations)	30
CalCol [Sub]	30
CalendarHide [Sub]	30
CalendarShow [Sub]	30
CalFormulaReplacement [Sub]	31
CheckForSheet [Sub]	31
CreateCalSht [Sub]	32
GroupCal [Sub]	34
MacroLinkRemover [Sub]	34
NextMonth [Sub]	35
NextYear [Sub]	35
PrevMonth [Sub]	35
PrevYear [Sub]	36
ReplaceCalendar [Sub]	36
SelectDay [Sub]	36
ShowSettings [Sub]	36
UnGroupCal [Sub]	36
Recurring_Macros	38
(Declarations)	38
Recurring_CheckAndCreate [Sub]	38
Recurring_Delete [Sub]	38
Recurring_FrequencyChange [Sub]	39
Recurring_Open [Sub]	39
Recurring_SaveUpdate [Sub]	40

```
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("D6:D12" )) Is Nothing And Range("C" & Target.Row).
8         Value <> Empty Then
9         Range("D6:D12" ).ClearContents
10        Target.Value = Chr(252)
11        Range("C6" ).Select
12    End If
13
14    'On Default Payment Terms Selection
15    If Not Intersect(Target, Range("H6:H23" )) Is Nothing And Range("F" & Target.Row).
16        Value <> Empty Then
17        Range("H6:H23" ).ClearContents
18        Target.Value = Chr(252)
19        Range("F6" ).Select
20    End If
21 End Sub
```

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Target.CountLarge > 2 Then Exit Sub
5     'Tab Selection
6     If Not Intersect(Target, Range("B2:E2" )) Is Nothing Then
7         Range("AA3" ).Value = Target.Column
8         Dashboard_TabChange 'Run Macro to switch tabs
9     End If
10
11     If Shapes("ViewInvBtn" ).Visible = True Then Shapes("ViewInvBtn" ).Visible =
12         msoFalse 'Hide View Inv. Button if Visible
13     If Not Intersect(Target, Range("B35:H9999" )) Is Nothing And Range("B" & Target.Row
14         ).Value <> Empty Then
15         Range("AA1" ).Value = Target.Row
16         Aging_ShowCustDetail
17     End If
18     If Not Intersect(Target, Range("J37:R99999" )) Is Nothing And Range("J" & Target.
19         Row).Value <> Empty And IsNumeric(Range("J" & Target.Row).Value) = True Then
20         Range("AA2" ).Value = Target.Row
21         With Shapes("ViewInvBtn" )
22             .Left = Range("S" & Target.Row).Left
23             .Top = Range("S" & Target.Row).Top
24             .Visible = msoCTrue
25         End With
26     Else
27         Range("AA2" ).ClearContents
28     End If
29 End Sub
```



```
1 Option Explicit
```

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

```
4     If Target.CountLarge > 1 Then Exit Sub
```

```
5
6     'On customer name change
```

```
7     If Not Intersect(Target, Range("G5" )) Is Nothing And Range("G5" ).Value <> Empty
```

```
8     Then
```

```
9         Dim CustRow As Long
```

```
10        If Range("B2" ).Value <> Empty Then 'Customer exists
```

```
11            CustRow = Range("B2" ).Value
```

```
12            Range("G6" ).Value = Customers.Range("C" & CustRow).Value 'Customer Address
```

```
13            Range("G7" ).Value = Customers.Range("D" & CustRow).Value _
```

```
14                & " " & Customers.Range("E" & CustRow).Value _
```

```
15                & ", " & Customers.Range("F" & CustRow).Value
```

```
16        Else 'Customer Not Found
```

```
17            Range("G6,G7" ).ClearContents 'Clear Address
```

```
18            If MsgBox("This customer does not exist. Would you like to add them" , vbYesNo
```

```
19                , "Customer Not Found" ) = vbYes Then
```

```
20                AddCustForm.Field1.Value = Range("G5" ).Value 'Customer Name
```

```
21                AddCustForm.Show
```

```
22            End If
```

```
23        End If
```

```
24    End If
```

```
25     'On Invoice Search # Change
```

```
26     If Not Intersect(Target, Range("L1" )) Is Nothing And Range("L1" ).Value <> Empty
```

```
27     Then
```

```
28         If Range("B4" ).Value = Empty Then
```

```
29             MsgBox "Please make sure to enter a correct invoice #"
```

```
30             Exit Sub
```

```
31         End If
```

```
32         Range("J1" ).Value = Range("L1" ).Value 'Add Inv. #
```

```
33         Range("L1" ).ClearContents
```

```
34         Invoice_Load 'Run Macro to Load invoice
```

```
35     End If
```

```
36     'On Change of Invoice Date or Terms but not on Invoice load
```

```
37     If Not Intersect(Target, Range("I5,I3" )) Is Nothing And Range("B6" ).Value = False
```

```
38     Then
```

```
39         If Range("I3" ).Value = Empty Or Range("I5" ).Value = Empty Then Exit Sub
```

```
40         Dim PmtTerm As String
```

```
41         Dim PmtRow As Long
```

```
42         PmtTerm = Range("I5" ).Value 'Payment Term
```

```
43         On Error Resume Next
```

```
44         PmtRow = Admin.Range("Terms" ).Find(PmtTerm, , xlValues, xlWhole).Row
```

```
45         On Error GoTo 0
```

```
46         If PmtRow <> 0 Then Range("I6" ).Value = Range("I3" ).Value + Admin.Range("G" &
```

```
47         PmtRow).Value 'Set Due Date
```

```
48     End If
```

```
49     'On Item ID Change (but not on Invoice Load)
```

```
50     If Not Intersect(Target, Range("C9:C31" )) Is Nothing And Range("B6" ).Value = False
```

```
51     Then
```

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

```
53             Dim ItemRow As Long
```

```
54             On Error Resume Next
```

```
55             ItemRow = Items.Range("Item_ID" ).Find(Target.Value, , xlValues, xlWhole).Row
```

```
1 2 3
```

```

1 2 3
'Item Row (if Found)
55 On Error GoTo 0
56 Range("F" & Target.Row).Validation.Delete 'Delete any validation if exists
57 If ItemRow <> 0 Then
58     Range("D" & Target.Row).Value = Items.Range("B" & ItemRow).Value 'Item
        Type
59     If Range("D" & Target.Row).Value = "Item" Then 'Item Type
60         Range("F" & Target.Row).Value = Items.Range("C" & ItemRow).Value
            'Item Name
61         Range("K" & Target.Row).Value = Items.Range("G" & ItemRow).Value
            'Item Cost
62     Else 'Service Type
63         Range("F" & Target.Row).ClearContents
64         Range("F" & Target.Row).Validation.Add xlValidateList, xlValidAlertStop
            , xlBetween, "=Tech_Name"
65     End If
66     Range("G" & Target.Row).Value = Items.Range("D" & ItemRow).Value 'Item
        Description
67     Range("H" & Target.Row).Value = Items.Range("E" & ItemRow).Value 'Item
        Def. Qty
68     Range("I" & Target.Row).Value = Items.Range("F" & ItemRow).Value 'Item
        Sales Price
69     End If
70 Else 'User Deleted Item - Clear Row
71     Range("D" & Target.Row & ":I" & Target.Row).ClearContents 'Clear Out Item
        Details
72     Range("K" & Target.Row).ClearContents 'Clear Out Item cost
73     If Range("B" & Target.Row).Value <> Empty Then 'DB Row Exists
74         Dim InvItemDBRow As Long
75         InvItemDBRow = Range("B" & Target.Row).Value
76         InvoiceItems.Range(InvItemDBRow & ":" & InvItemDBRow).EntireRow.Delete
            'Delete DB Row
77         Range("B" & Target.Row).ClearContents 'Clear Out DB Row
78     End If
79 End If
80 End If
81
82 'On Change of Technician, update/change Hourly Billing Rate & Cost (Not on INvoice
Load)
83 If Not Intersect(Target, Range("F9:F31" )) Is Nothing And Range("B6" ).Value = False
And Range("F" & Target.Row).Value <> Empty Then
84     Dim TechRow As Long
85     On Error Resume Next
86     TechRow = Techs.Range("Tech_Name" ).Find(Target.Value, , xlValues, xlWhole).Row
87     On Error GoTo 0
88     If TechRow <> 0 Then
89         Range("I" & Target.Row).Value = Techs.Range("E" & TechRow).Value 'Billing
            Rate
90         Range("K" & Target.Row).Value = Techs.Range("K" & TechRow).Value
            'Calculated Hourly Cost
91     End If
92 End If
93 End Sub
94
95 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
96     If Target.CountLarge > 1 Then Exit Sub
97     'Check To Make sure calendar exists, if not, copy from calpop up
98     Dim Cal As Shape
99     On Error Resume Next
100    Set Cal = Shapes("Calendar" )

```

```
1
101 On Error GoTo 0
102 If Cal Is Nothing Then ReplaceCalendar 'Run Macro to Replace Calendar
103
104 If Not Intersect(Target, Range("J3,E9:E31" )) Is Nothing Then 'Check For Sheet and
Calendar
105     CheckForSheet 'Macro to check for sheet if not create
106     CalendarShow
107 Else
108     CheckForSheet
109     CalendarHide
110 End If
111 End Sub
```

1	Option Explicit
2	

1 Option Explicit

2

1

Option Explicit

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     'On Invoice #, Add Customer Details
5     If Not Intersect(Target, Range("E3" )) Is Nothing And Range("E3" ).Value <> Empty
6         Then
7             Dim CustRow As Long
8             If Range("B6" ).Value <> Empty Then 'Customer Found
9                 CustRow = Range("B6" ).Value
10                Range("D5" ).Value = Customers.Range("B" & CustRow).Value 'Customer Name
11                Range("D6" ).Value = Customers.Range("C" & CustRow).Value 'Customer Address
12                Range("D7" ).Value = Customers.Range("D" & CustRow).Value _
13                    & " " & Customers.Range("E" & CustRow).Value _
14                    & ", " & Customers.Range("F" & CustRow).Value 'City State Zip
15                If Range("D5" ).Value <> Empty Then Payment_CustomerPmntsRefresh
16            End If
17        End If
18    End Sub
19
20 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
21     If Target.CountLarge > 1 Then Exit Sub
22     'Check To Make sure calendar exists, if not, copy from calpop up
23     Dim Cal As Shape
24     On Error Resume Next
25     Set Cal = Shapes("Calendar" )
26     On Error GoTo 0
27     If Cal Is Nothing Then ReplaceCalendar 'Run Macro to Replace Calendar
28
29     If Not Intersect(Target, Range("H3" )) Is Nothing Then 'Check For Sheet and
30         Calendar
31         CheckForSheet 'Macro to check for sheet if not create
32         CalendarShow
33     Else
34         CheckForSheet
35         CalendarHide
36     End If
37 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

```
1 Option Explicit
2
3
4
5
6
7
8
9 Private Sub CancelBtn_Click()
10     AddCustForm.Hide
11 End Sub
12
13 Private Sub SaveBtn_Click()
14     Dim CustomerFld As Control
15     Dim CustRow As Long, CustCol As Long
16     If Me.Field1.Value = Empty Then
17         MsgBox "Please make sure to save a Customer Name before saving"
18         Exit Sub
19     End If
20     With Customers
21         CustRow = .Range("A99999").End(xlUp).Row + 1 'First avail Row
22         On Error Resume Next
23         .Range("A" & CustRow).Value = Application.WorksheetFunction.Max(.Range(
24             "Cust_ID" )) + 1
25         On Error GoTo 0
26         If .Range("A" & CustRow).Value = Empty Then .Range("A" & CustRow).Value = 1
27         'Set First Cust ID to 1
28         For CustCol = 2 To 8
29             Set CustomerFld = Me.Controls("Field" & CustCol - 1)
30             .Cells(CustRow, CustCol).Value = CustomerFld.Value
31         Next CustCol
32         AddCustForm.Hide
33         Invoice.Range("G5").Value = Field1.Value 'Set Customer Name
34     End With
35 End Sub
```

```
1 Option Explicit
2
3
4
5 Private Sub CancelBtn_Click()
6     Me.Hide
7 End Sub
8
9 Private Sub ClientNmLbl_Click()
10    Recurring_Delete
11 End Sub
12
13 Private Sub DeleteBtn_Click()
14    Recurring_Delete
15 End Sub
16
17 Private Sub RecField5_Change()
18    Recurring_FrequencyChange
19 End Sub
20
21 Private Sub RecField6_AfterUpdate()
22    'Recurring_FrequencyChange
23 End Sub
24
25 Private Sub RecField6_Change()
26    Recurring_FrequencyChange
27 End Sub
28
29 Private Sub SaveBtn_Click()
30    Recurring_SaveUpdate
31 End Sub
32
```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, SelRow As Long, SelCol As Long, DetailNumb
  As Long
3
4 Sub Aging_GoToInvoice()
5     With Dashboard
6         SelRow = .Range("AA2" ).Value 'Selected Row
7         If SelRow = 0 Then Exit Sub
8         If .Range("J" & SelRow).Value = "" Then Exit Sub
9         Invoice.Activate
10        Invoice.Range("L1" ).Value = .Range("J" & SelRow).Value 'set Invoice #
11    End With
12 End Sub
13
14 Sub Aging_Refresh()
15     Dashboard.Range("B35:R499" ).ClearContents 'Clear Previous Results
16     With Invoicelist
17         'Clear Prior Results
18         .Range("AB3:AJ9999" ).ClearContents
19         LastRow = .Range("A99999" ).End(xlUp).Row
20         If LastRow < 3 Then Exit Sub
21         .Range("H3:J" & LastRow).Formula = .Range("H1:J1" ).Formula 'Bring Down Total
22         Paid & Days Overdue Formulas
23         .Range("A2:D" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
24         "L1:L2" ), CopyToRange:=.Range("P2" ), Unique:=True
25         LastResultRow = .Range("P99999" ).End(xlUp).Row
26         If LastResultRow < 3 Then Exit Sub
27         .Range("Q3:V" & LastResultRow).Formula = .Range("Q1:W1" ).Formula
28         Dashboard.Range("B35:H" & LastResultRow + 32).Value = .Range("P3:V" &
29         LastResultRow).Value 'Bring over Aging Data
30     End With
31 End Sub
32
33 Sub Aging_ShowCustDetail()
34     Dashboard.Range("J34:R499" ).ClearContents 'Clear Previous Results
35     SelRow = Dashboard.Range("AA1" ).Value 'Set Selected Row
36     With Invoicelist
37         'Clear Prior Results
38         .Range("AB3:AJ9999" ).ClearContents
39         LastRow = .Range("A99999" ).End(xlUp).Row
40         If LastRow < 3 Then Exit Sub
41         .Range("H3:J" & LastRow).Formula = .Range("H1:J1" ).Formula 'Bring Down Total
42         Paid & Days Overdue Formulas
43         .Range("A2:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
44         "Y1:Z2" ), CopyToRange:=.Range("AB2:AJ2" ), Unique:=True
45         LastResultRow = .Range("AB99999" ).End(xlUp).Row
46         If LastResultRow < 3 Then Exit Sub
47         Dashboard.Range("J" & SelRow & ":R" & SelRow + LastResultRow - 1).Value = .
48         Range("AB1:AJ" & LastResultRow).Value 'Bring over Customer Details
49         Dashboard.Range("J" & SelRow & ":R" & SelRow).HorizontalAlignment =
50         xlCenterAcrossSelection
51     End With
52 End Sub
53
54 Sub AgingDetail_Refresh()
55     Dashboard.Range("B507:J9999" ).ClearContents 'Clear Existing Data
56     With Invoicelist
57         'Clear Prior Results

```

1 2


```

1 2
53 .Range("AB3:AJ9999" ).ClearContents
54 LastRow = .Range("A9999" ).End(xlUp).Row
55 If LastRow < 3 Then Exit Sub
56 .Range("H3:J" & LastRow).Formula = .Range("H1:J1" ).Formula 'Bring Down Total
    Paid & Days Overdue Formulas
57 .Range("A2:ND" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "AL1:AM2" ), CopyToRange:=.Range("AB2:AJ2" ), Unique:=True
58 LastResultRow = .Range("AB99999" ).End(xlUp).Row
59 If LastResultRow < 3 Then Exit Sub
60 Dashboard.Range("B507:J" & LastResultRow + 504).Value = .Range("AB3:AJ" &
    LastResultRow).Value 'Bring over Aging Data
61 End With
62
63 End Sub
64
65 Sub Dashboard_Refresh()
66 'Get Current Data
67 With Invoicelist
68 'Clear Prior Results
69 .Range("AB3:AJ9999" ).ClearContents
70 LastRow = .Range("A9999" ).End(xlUp).Row
71 If LastRow < 3 Then Exit Sub
72 .Range("H3:J" & LastRow).Formula = .Range("H1:J1" ).Formula 'Bring Down Total
    Paid & Days Overdue Formulas
73 .Range("A2:D" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "L1:L2" ), CopyToRange:=.Range("P2" ), Unique:=True
74 LastResultRow = .Range("P99999" ).End(xlUp).Row
75 If LastResultRow < 3 Then Exit Sub
76 .Range("Q3:W" & LastResultRow).Formula = .Range("Q1:W1" ).Formula
77 End With
78 End Sub
79
80 Sub Dashboard_SelectAgingDetails()
81 DetailNumb = Replace(Application.Caller, "Aging" , "" )
82 Dashboard.Range("AA504" ).Value = DetailNumb 'Set Detail level #
83 Dashboard.Range("E2" ).Select 'Select to trigger macro
84 End Sub
85
86 Sub Dashboard_TabChange()
87 Application.ScreenUpdating = False
88 With Dashboard
89 SelCol = .Range("AA3" ).Value 'Selected Column
90 'Hide All Shapes & Graphs
91 .Range("4:1004" ).EntireRow.Hidden = True
92 .Shapes("DashGrp" ).Visible = msoFalse 'Hide Dashboard Group
93 .Shapes("DetailGrp" ).Visible = msoFalse 'Hide Details Group
94 Select Case SelCol
95 Case Is = 2 'Dashboard
96 .Range("4:32" ).EntireRow.Hidden = False
97 .Shapes("DashGrp" ).Visible = msoCTrue 'Display Dashboard Graphs/Buttons
98 Dashboard_Refresh
99 Case Is = 3 'Aging Summary
100 .Range("33:502" ).EntireRow.Hidden = False
101 Aging_Refresh 'Run Macro To Refresh Aging
102 Case Is = 5 'Aging Detail
103 .Range("503:1004" ).EntireRow.Hidden = False
104 .Shapes("DetailGrp" ).Visible = msoCTrue 'Show Detail group
105 AgingDetail_Refresh 'Run Macro to refresh aging detail
106 End Select
107 .Range("A1" ).Select

```

```
108 | 1 2 End With  
109 | Application.ScreenUpdating = True  
110 | End Sub
```

```

1 Option Explicit
2 Dim InvRow As Long, InvItemRow As Long, LastRow As Long, LastItemRow As Long
3 Dim ResultRow As Long, LastResultRow As Long, ItemRow As Long, TermRow As Long,
  StatusRow As Long
4
5 Sub Customer_AddNew()
6     Unload AddCustForm
7     AddCustForm.Show
8 End Sub
9
10 Sub Invoice_Delete()
11     If MsgBox("Are you sure you want to delete this Invoice?" , vbYesNo, "Delete
  Appointment" ) = vbNo Then Exit Sub
12     With Invoice
13         If .Range("B3" ).Value = Empty Then GoTo NotSaved
14         InvRow = .Range("B3" ).Value 'Order Row
15         InvoiceList.Range(InvRow & ":" & InvRow).EntireRow.Delete
16         'Remove Invoice Items
17         With InvoiceItems
18             LastRow = .Range("A9999" ).End(xlUp).Row
19             If LastRow < 3 Then GoTo NotSaved
20             .Range("A2:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
  "M2:M3" ), CopyToRange:=.Range("P2:Y2" ), Unique:=True
21             LastResultRow = .Range("P9999" ).End(xlUp).Row
22             If LastResultRow < 3 Then GoTo NotSaved
23             If LastResultRow < 4 Then GoTo SingleRow
24             'Sort based on descending rows
25             With .Sort
26                 .SortFields.Clear
27                 .SortFields.Add Key:=InvoiceItems.Range("P3" ), SortOn:=xlSortOnValues,
  Order:=xlDescending, DataOption:=xlSortNormal
28                 .SetRange InvoiceItems.Range("P3:Y" & LastResultRow)
29                 .Apply
30             End With
31             SingleRow:
32             For ResultRow = 3 To LastResultRow
33                 InvItemRow = .Range("P" & ResultRow).Value 'Invoice Item DB Row
34                 If InvItemRow > 3 Then .Range(InvItemRow & ":" & InvItemRow).EntireRow.
  Delete 'Don't remove 1st Row
35             Next ResultRow
36         End With
37         NotSaved:
38         Invoice_New 'Clear Out All Invoice Fields
39     End With
40 End Sub
41
42 Sub Invoice_Load()
43     With Invoice
44         If .Range("B3" ).Value = Empty Then
45             MsgBox "Please entere a correct invoice #"
46             Exit Sub
47         End If
48         InvRow = .Range("B3" ).Value 'Invoice Row
49         .Range("B6" ).Value = True 'Set Inv. Load to true
50         .Range("B7,I3:J6,G5:G7,B9:I31,K9:K31" ).ClearContents
51         .Range("I3" ).Value = InvoiceList.Range("B" & InvRow).Value 'Inv. Date
52         .Range("G5" ).Value = InvoiceList.Range("C" & InvRow).Value 'Customer
53         .Range("I4" ).Value = InvoiceList.Range("D" & InvRow).Value 'Inv. Status
54         .Range("I5" ).Value = InvoiceList.Range("E" & InvRow).Value 'Terms
55         .Range("I6" ).Value = InvoiceList.Range("F" & InvRow).Value 'Due Date

```

1 2

```

1 2
56 .Range("B7" ).Value = InvoiceList.Range("K" & InvRow).Value 'Recurring ID (if
any)
57
58 With InvoiceItems
59 LastRow = .Range("A99999" ).End(xlUp).Row
60 If LastRow < 3 Then GoTo NoItems
61 .Range("A2:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"M2:M3" ), CopyToRange:=.Range("P2:Y2" ), Unique:=True
62 LastResultRow = .Range("P99999" ).End(xlUp).Row
63 If LastResultRow < 3 Then GoTo NoItems
64 For ResultRow = 3 To LastResultRow
65 InvItemRow = .Range("Y" & ResultRow).Value 'Get Invoice Row
66 Invoice.Range("B" & InvItemRow & ":I" & InvItemRow).Value = .Range("P" &
ResultRow & ":W" & ResultRow).Value 'Item Details
67 Invoice.Range("K" & InvItemRow).Value = InvoiceItems.Range("X" &
ResultRow).Value 'Item Cost
68 Next ResultRow
69 End With
70 NoItems:
71 .Range("B6" ).Value = False 'Set inv. Load to false
72 End With
73 End Sub
74
75 Sub Invoice_New()
76 With Invoice
77 .Range("B6" ).Value = True 'Set Inv. Load to true
78 .Range("B7,I3:J6,G5:G7,B9:I31,K9:K31" ).ClearContents
79 .Range("J1" ).Value = .Range("B5" ).Value 'Set Next invoice #
80 .Range("I3" ).Value = Date 'Set current Date
81 .Range("B6" ).Value = False 'Set inv. Load to false
82 On Error Resume Next
83 TermRow = Admin.Range("H6:H23" ).Find(Chr(252), , xlValues, xlWhole).Row
84 On Error GoTo 0
85 If TermRow <> 0 Then .Range("I5" ).Value = Admin.Range("F" & TermRow).Value
'Set Default Term
86 On Error Resume Next
87 StatusRow = Admin.Range("D6:D12" ).Find(Chr(252), , xlValues, xlWhole).Row
88 On Error GoTo 0
89 If StatusRow <> 0 Then .Range("I4" ).Value = Admin.Range("C" & StatusRow).Value
'Set Default Status
90 End With
91 End Sub
92
93
94 Sub Invoice_Print()
95 Invoice.PrintOut , , , True, , , False
96 End Sub
97
98 Sub Invoice_SaveAsPDF()
99 Dim FilePath As String
100 Invoice_SaveUpdate 'Save invoice
101 FilePath = ThisWorkbook.Path & "\" & Invoice.Range("G5" ).Value & "_" & Invoice.
Range("J1" ).Value 'File Path
102 If Dir(FilePath, vbDirectory) <> "" Then Kill (FilePath)
103 Invoice.ExportAsFixedFormat xlTypePDF, FilePath, , , False, , , True
104 End Sub
105 Sub Invoice_SavedMsg()
106 With Invoice.Shapes("InvSavedMsg" )
107 Dim i As Long, Delay As Double, StartTime As Double
108 .Visible = msoCTrue

```

```

1 2
109 For i = 1 To 150
110     .Fill.Transparency = i / 150
111     Delay = 0.009
112     StartTime = Timer
113     Do
114         DoEvents
115     Loop While Timer - StartTime < Delay
116 Next i
117 .Visible = msoFalse
118 End With
119 End Sub
120 Sub Invoice_SaveUpdate()
121     With Invoice
122         If .Range("G5").Value = Empty Then
123             MsgBox "Please make sure to add a customer before saving invoice"
124             Exit Sub
125         End If
126
127         'Deterermine New Invoice/Existing Invoice
128         If .Range("B3").Value = Empty Then 'new Invoice
129             InvRow = InvoiceList.Range("A9999").End(xlUp).Row + 1 'First Avail Row
130             .Range("J1").Value = .Range("B5").Value 'Next Inv. #
131             InvoiceList.Range("A" & InvRow).Value = .Range("B5").Value 'Next Inv. #
132         Else 'Existing Invoice
133             InvRow = .Range("B3").Value 'Invoice Row
134         End If
135         InvoiceList.Range("B" & InvRow).Value = .Range("I3").Value 'Date
136         InvoiceList.Range("C" & InvRow).Value = .Range("G5").Value 'Customer
137         InvoiceList.Range("D" & InvRow).Value = .Range("I4").Value 'Status
138         InvoiceList.Range("E" & InvRow).Value = .Range("I5").Value 'Terms
139         InvoiceList.Range("F" & InvRow).Value = .Range("I6").Value 'Due Date
140         InvoiceList.Range("G" & InvRow).Value = .Range("J34").Value 'Invoice Total
141         InvoiceList.Range("K" & InvRow).Value = .Range("B7").Value 'Recurring ID (if
any)
142
143         'Add/Update Invoice Items
144         LastItemRow = .Range("C31").End(xlUp).Row 'Last Invoice Row
145         If LastItemRow < 9 Then GoTo NoItems
146         For ItemRow = 9 To LastItemRow
147             If .Range("B" & ItemRow).Value <> Empty Then 'DB Row Exists
148                 InvItemRow = .Range("B" & ItemRow).Value 'Inv. Item DB Row
149             Else 'New DB ItemRow
150                 InvItemRow = InvoiceItems.Range("A9999").End(xlUp).Row + 1 'First avail
Row
151                 InvoiceItems.Range("A" & InvItemRow).Value = .Range("J1").Value 'Inv ID
152                 .Range("B" & ItemRow).Value = InvItemRow 'Add DB Row
153                 InvoiceItems.Range("K" & InvItemRow).Value = "=Row()" 'Add Row Formula
154             End If
155             InvoiceItems.Range("B" & InvItemRow & ":H" & InvItemRow).Value = .Range("C"
& ItemRow & ":I" & ItemRow).Value 'Bring over item Values
156             InvoiceItems.Range("I" & InvItemRow).Value = .Range("K" & ItemRow).Value
'Item Cost
157             InvoiceItems.Range("J" & InvItemRow).Value = ItemRow 'Invoice Row
158         Next ItemRow
159
160         NoItems:
161     End With
162     Invoice_SavedMsg 'Run Invoice Fade Out Message
163 End Sub

```

```

1 Option Explicit
2 Dim PmntRow As Long, PmntID As Long, LastRow As Long, LastResultRow As Long, InvRow As
   Long
3 Sub Payment_CustomerPmntsRefresh()
4     Payments.Range("D13:H999").ClearContents 'Clear existing Data
5     With PmntsDB
6         LastRow = .Range("A99999").End(xlUp).Row
7         If LastRow < 3 Then Exit Sub
8         .Range("A2:F" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
           "K1:K2"), CopyToRange:=.Range("M2:P2"), Unique:=True
9         LastResultRow = .Range("M99999").End(xlUp).Row
10        If LastResultRow < 3 Then Exit Sub
11        If LastResultRow < 4 Then GoTo SkipSort
12        With .Sort
13            .SortFields.Clear
14            .SortFields.Add Key:=PmntsDB.Range("M3"), SortOn:=xlSortOnValues, Order:=
           xlDescending, DataOption:=xlSortNormal
15            .SetRange PmntsDB.Range("M3:P" & LastResultRow)
16            .Apply
17        End With
18        SkipSort:
19        Payments.Range("D13:G" & LastResultRow + 10).Value = .Range("M3:P" &
           LastResultRow).Value
20    End With
21 End Sub
22
23
24 Sub Payment_Delete()
25     If MsgBox("Are you sure you want to delete this Payment?", vbYesNo, "Delete
       Payment") = vbNo Then Exit Sub
26     With Payments
27         If .Range("B3").Value = Empty Then GoTo NotSaved
28         PmntRow = .Range("B3").Value 'Payment Row
29         PmntsDB.Range(PmntRow & ":" & PmntRow).EntireRow.Delete 'Delete Payment Row
30         NotSaved:
31         Payment_New
32     End With
33 End Sub
34
35 Sub Payment_Load()
36     With Payments
37         .Range("B7,E3,H7,D5:D7,E9:H9,D13:H999").ClearContents
38         If .Range("B3").Value = Empty Then
39             MsgBox "Please select a correct payment"
40             Exit Sub
41         End If
42         PmntRow = .Range("B3").Value 'Payment Row
43         .Range("H3").Value = PmntsDB.Range("B" & PmntRow).Value 'Date
44         .Range("D5").Value = PmntsDB.Range("C" & PmntRow).Value 'Customer
45         .Range("E3").Value = PmntsDB.Range("D" & PmntRow).Value 'Invoice #
46         .Range("H7").Value = PmntsDB.Range("E" & PmntRow).Value 'Amount
47         .Range("E9").Value = PmntsDB.Range("F" & PmntRow).Value 'Notes
48         .Range("B7").Value = PmntsDB.Range("G" & PmntRow).Value 'Recurring ID (If any)
49         If .Range("B6").Value <> "" Then Payment_CustomerPmntsRefresh 'Load Previous
           Customer Payments
50     End With
51 End Sub
52 Sub Payment_New()
53     Payments.Range("B2,B7,E3,H7,D5:D7,E9:H9,D13:H999").ClearContents

```

1

```

1
54 Payments.Range("E3").Select
55 End Sub
56
57 Sub Payment_Next()
58 With Payments
59 PmntID = .Range("B2").Value 'Payment ID
60 If PmntID = 0 Then 'No Current ID
61     If .Range("B4").Value = 1 Then 'No Saved Payments
62         MsgBox "Please save any Payments first before navigating to previously
        saved"
63         Exit Sub
64     End If
65     .Range("B2").Value = 1 'Set Pmnt. ID to the first one created
66     Payment_Load
67     Exit Sub
68 End If
69 If PmntID = .Range("B4").Value - 1 Then
70     MsgBox "You are already at the last Payment created"
71     Exit Sub
72 End If
73 .Range("B2").Value = .Range("B2").Value + 1 'Set next Payment ID
74 Payment_Load
75 End With
76 End Sub
77
78 Sub Payment_Prev()
79 With Payments
80 PmntID = .Range("B2").Value 'Payment ID
81 If PmntID = 0 Then 'No Current ID
82     If .Range("B4").Value = 1 Then 'No Saved Payments
83         MsgBox "Please save any Payments first before navigating to previously
        saved"
84         Exit Sub
85     End If
86     .Range("B2").Value = .Range("B4").Value - 1 'Set Pmnt. ID to the last one
        created
87     Payment_Load
88     Exit Sub
89 End If
90 If PmntID = 1 Then
91     MsgBox "You are already at the first Payment created"
92     Exit Sub
93 End If
94 .Range("B2").Value = .Range("B2").Value - 1 'Set Previous Payment ID
95 Payment_Load
96 End With
97 End Sub
98 Sub Payment_SavedMsg()
99 With Payments.Shapes("PmntSavedMsg")
100     Dim i As Long, Delay As Double, StartTime As Double
101     .Visible = msoCTrue
102     For i = 1 To 150
103         .Fill.Transparency = i / 150
104         Delay = 0.009
105         StartTime = Timer
106         Do
107             DoEvents
108             Loop While Timer - StartTime < Delay
109         Next i
110     .Visible = msoFalse

```

1 2

```

111 End With
112 End Sub
113 Sub Payment_SaveUpdate()
114 With Payments
115     If .Range("B5").Value = Empty Then 'Incorrect / Missing Invoice #
116         MsgBox "Please select a correct Invoice #"
117         Exit Sub
118     End If
119     InvRow = .Range("B5").Value 'Invoice Row
120     If .Range("H3").Value = Empty Or .Range("H7").Value = Empty Then 'Empty Fields
121         MsgBox "Please make sure to add in a Payment Date and Payment Amount"
122         Exit Sub
123     End If
124
125     If .Range("H8").Value < 0 Then
126         If MsgBox("The Payment Amount is above the Invoice Balance. Are you sure you
127             want to continue?", vbYesNo, "Payment Amount Issue") = vbNo Then Exit Sub
128     End If
129
130     If .Range("B3").Value = Empty Then 'New Payment
131         PmntRow = PmntsDB.Range("A99999").End(xlUp).Row + 1
132         .Range("B2").Value = .Range("B4").Value 'Next Pment ID
133         PmntsDB.Range("A" & PmntRow).Value = .Range("B4").Value 'Next Payment ID
134     Else 'Existing Payment
135         PmntRow = .Range("B3").Value 'Payment Row
136     End If
137     PmntsDB.Range("B" & PmntRow).Value = .Range("H3").Value 'Date
138     PmntsDB.Range("C" & PmntRow).Value = .Range("D5").Value 'Customer
139     PmntsDB.Range("D" & PmntRow).Value = .Range("E3").Value 'Invoice #
140     PmntsDB.Range("E" & PmntRow).Value = .Range("H7").Value 'Amount
141     PmntsDB.Range("F" & PmntRow).Value = .Range("E9").Value 'Notes
142     PmntsDB.Range("G" & PmntRow).Value = .Range("B7").Value 'Recurring ID (If any)
143
144     'Update invoice Paid Status
145     If .Range("H8").Value = 0 Then 'Paid
146         InvoiceList.Range("D" & InvRow).Value = Admin.Range("C10").Value 'Fully
147         Paid
148     Else ' Partial Paid
149         InvoiceList.Range("D" & InvRow).Value = Admin.Range("C9").Value 'Partially
150         Paid
151     End If
152 End With
153 Payment_CustomerPmntsRefresh 'Refresh Customer Payments
154 Payment_SavedMsg 'Run Fade out message
155 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 = Selection
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("Calendar").GroupItems("NextTri").OnAction, "!") <> 0 Or
   InStr(.Shapes("1Day").DrawingObject.Formula, "]" ) <> 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 With ActiveSheet.Shapes(DayName)
65     .Fill.ForeColor.RGB = RGB(252, 213, 180)
66     .TextFrame2.TextRange.Font.Bold = msoTrue
67 End With
68 On Error GoTo NoCal
69 .Shapes("Calendar").Visible = msoCTrue
70 .Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3", "CalCol4",
   "CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible = False '
71 .Shapes("Calendar").Left = SelCell.Left
72 .Shapes("Calendar").Placement = xlMove
73 .Shapes("Calendar").Top = SelCell.Offset(1, 0).Top
74 If Sheets("CalPopUp").Range("A6").Value > 0 Then
75     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
   "42Day")).Visible = True
76 Else:
77     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
   "42Day")).Visible = False
78 End If
79 Sheets("CalPopUp").Range("A7").Value = SelCell.Address
80 ActiveCell.Select
81 End With
82 Exit Sub
83 NoCal:
84 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
   calendar over from another sheet and paste into this sheet"
85 End Sub
86 Sub CalFormulaReplacement()
87     With ActiveSheet
88         Dim DayNum, ColNum, RowNum As Long
89         Dim Shp As Shape
90         ColNum = 2
91         RowNum = 1
92         For DayNum = 1 To 42
93             .Shapes(DayNum & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(RowNum
   , ColNum).Address
94             ColNum = ColNum + 1
95             If ColNum = 9 Then
96                 ColNum = 2
97                 RowNum = RowNum + 1
98             End If
99         Next DayNum
100        .Shapes("Month").DrawingObject.Formula = "=CalPopUp!$A$4"
101        .Shapes("Year").DrawingObject.Formula = "=CalPopUp!$A$2"
102    End With
103 End Sub
104
105 Sub CheckForSheet()
106     'Checks for existance of Calendar Pop-up Worksheet
107     Dim ws As Worksheet
108     On Error GoTo CreateWS
109     Set ws = ActiveWorkbook.Sheets("CalPopUp")
110     Exit Sub

```

```

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

```

1 2

```

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

```

```

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

```

1 2

```

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

```

```

1
307 With Sheets("CalPopUp" )
308     If .Range("A3" ).Value = 1 Then
309         .Range("A3" ).Value = 12
310         .Range("A2" ).Value = .Range("A2" ).Value - 1
311     Else:
312         .Range("A3" ).Value = .Range("A3" ).Value - 1
313     End If
314     If .Range("A6" ).Value > 0 Then
315         ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
316             "40Day" , "41Day" , "42Day" )).Visible = True
317     Else:
318         ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
319             "40Day" , "41Day" , "42Day" )).Visible = False
320     End If
321 End With
322 End Sub

323 Sub PrevYear()
324     ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
325         "CalPopUp" ).Range("A2" ).Value - 1
326 End Sub

327 Sub ReplaceCalendar() 'Shape Deleted
328     CalPopUp.Shapes("Calendar" ).Copy 'Copy From Developers sheet
329     ActiveCell.Select 'Select the active Cell
330     ActiveSheet.Paste 'Paste in Calendar
331 End Sub

332 .....
333 ''''Select Day Of The Month
334 .....

335 Sub SelectDay()
336     Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
337     DayNumb = Replace(Application.Caller, "Day" , "" )
338     RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
339     ColNumb = DayNumb Mod 7 + 1
340     If ColNumb = 1 Then ColNumb = 8
341     'On Error Resume Next
342     If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
343     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
344         ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
345     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
346     ActiveCell.Offset(0, 1).Select
347 End Sub

348 Sub ShowSettings()
349     'Show or Hide Calendar Settings Panel
350     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
351         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
352             "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
353         = False
354     Else:
355         ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
356             "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
357         = True
358     End If
359 End Sub

360 Sub UnGroupCal()
361     On Error Resume Next
362     ActiveSheet.Shapes("Calendar" ).Ungroup
363     ActiveSheet.Shapes("NextMonth" ).Ungroup

```

```
1  
359 ActiveSheet.Shapes("PrevMonth").Ungroup  
360 On Error GoTo 0  
361 End Sub
```



```

1 Option Explicit
2 Dim RecRow As Long, RecCol As Long, TransID As Long, RecID As Long, RecFreq As Long
3 Dim LastRow As Long, LastResultRow As Long, ResultRow As Long
4 Dim CustNm As String, TransType As String, FreqType As String
5 Dim NextTransDt As Date, TransDt As Date
6 Dim RecurFld As Control
7
8 Sub Recurring_CheckAndCreate()
9     With RecurDB
10        LastRow = .Range("A99999").End(xlUp).Row
11        If LastRow < 4 Then Exit Sub
12        .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
13            "J2:J3" ), CopyToRange:=.Range("L2:R2" ), Unique:=True
14        LastResultRow = .Range("L99999").End(xlUp).Row
15        If LastResultRow < 3 Then Exit Sub
16        For ResultRow = 3 To LastResultRow
17            RecID = .Range("L" & ResultRow).Value 'Recurring ID
18            TransType = .Range("M" & ResultRow).Value 'Trans Type
19            RecFreq = .Range("P" & ResultRow).Value 'Recurring Frequency
20            FreqType = .Range("Q" & ResultRow).Value 'Frequency Type
21            NextTransDt = .Range("R" & ResultRow).Value 'Next Trans Date
22            If TransType = "Invoice" Then
23                Invoice.Range("J1").Value = .Range("N" & ResultRow).Value 'Set Invoice
24                Invoice_Load 'Run Macro to load invoice
25                Invoice.Range("B7,J1, B9:B31").ClearContents 'Clear Invoice after Invoice
26                Invoice_SaveUpdate 'Save New Invoice
27            Else 'Payments
28                Payments.Range("B2").Value = .Range("N" & ResultRow).Value 'Set Payment
29                Payment_Load 'Run Macro To load Payment
30                Payments.Range("B2,B7").ClearContents 'Clear Payment ID
31                Payments.Range("H3").Value = NextTransDt 'Set Next Transaction Date
32                Payment_SaveUpdate 'Save New Payment
33            End If
34
35            'Update Next Date
36            Select Case FreqType
37            Case Is = "Day(s)"
38                NextTransDt = DateAdd("d", RecFreq, NextTransDt)
39            Case Is = "Week(s)"
40                NextTransDt = DateAdd("ww", RecFreq, NextTransDt)
41            Case Is = "Month(s)"
42                NextTransDt = DateAdd("m", RecFreq, NextTransDt)
43            Case Is = "Year(s)"
44                NextTransDt = DateAdd("d", RecFreq, NextTransDt)
45            End Select
46            On Error Resume Next
47            RecRow = RecurDB.Range("Rec_ID").Find(RecID, , xlValues, xlWhole).Row
48            On Error GoTo 0
49            If RecRow <> 0 Then .Range("G" & RecRow).Value = NextTransDt 'Update Next
50                Transaction Date
51        Next ResultRow
52    End With
53 End Sub
54
55

```

```

56 Sub Recurring_Delete()
57   If MsgBox("Are you sure you want to delete this recurring item?" , vbYesNo, "Delete
58     Recurring Item" ) = vbNo Then Exit Sub
59   With Recurring
60     RecID = .RecField1.Value 'RecID
61     On Error Resume Next
62     RecRow = RecurDB.Range("Rec_ID" ).Find(RecID, , xlValues, xlWhole).Row
63     On Error GoTo 0
64     If RecRow = 0 Then GoTo NotSaved
65     RecurDB.Range(RecRow & ":" & RecRow).EntireRow.Delete
66     ActiveSheet.Range("B7" ).ClearContents
67     NotSaved:
68     .Hide
69   End With
70 End Sub

71 Sub Recurring_FrequencyChange()
72   'Update Next Date On Frequency change
73   With Recurring
74     If .RecField5.Value = "" Or .RecField6.Value = Empty Then Exit Sub 'Exit on
75       either Field empty
76     If IsNumeric(.RecField5.Value) = False Then
77       MsgBox "Please make sure the Frequency Qty is a whole #"
78     End If
79     RecFreq = .RecField5.Value 'Recurring Frequency
80     FreqType = .RecField6.Value 'Frequency Type
81     Select Case FreqType
82     Case Is = "Day(s)"
83       NextTransDt = DateAdd("d" , RecFreq, Date)
84     Case Is = "Week(s)"
85       NextTransDt = DateAdd("ww" , RecFreq, Date)
86     Case Is = "Month(s)"
87       NextTransDt = DateAdd("m" , RecFreq, Date)
88     Case Is = "Year(s)"
89       NextTransDt = DateAdd("d" , RecFreq, Date)
90     End Select
91     .RecField7.Value = NextTransDt
92   End With
93 End Sub

94
95 End Sub

96
97 Sub Recurring_Open()
98   With ActiveSheet
99     If .Name = "Invoice" Then TransType = "Invoice" Else TransType = "Payment"
100    If TransType = "Invoice" Then
101      If Invoice.Range("B3" ).Value = "" Then
102        MsgBox "Please make sure to save this Invoice before making it recurring"
103        Exit Sub
104      End If
105      CustNm = .Range("G5" ).Value 'Customer Nmae
106      TransID = .Range("J1" ).Value ' Invoice ID
107    Else 'Recurring Payment
108      If Payments.Range("B3" ).Value = Empty Then
109        MsgBox "Please make sure to save this Payment before making it recurring"
110        Exit Sub
111      End If
112      CustNm = .Range("D5" ).Value 'Customer Name
113      TransID = .Range("B2" ).Value 'Payment ID

```

1 2 3

```

114 End If
115
116 RecID = .Range("B7").Value 'Recurring ID
117 If RecID = 0 Then 'New Recurring Transaction
118     On Error Resume Next
119     RecID = Application.WorksheetFunction.Max([Rec_ID]) + 1
120     On Error GoTo 0
121     If RecID = 0 Then RecID = 1
122     RecRow = RecurDB.Range("A9999").End(xlUp).Row + 1 'First AvailRow
123 Else: 'Existing Recurring
124     On Error Resume Next
125     RecRow = RecurDB.Range("Rec_ID").Find(RecID, , xlValues, xlWhole).Row
126     On Error GoTo 0
127     'If RecRow = 0 Then RecRow = 4 'Set First Reurring ID
128 End If
129 With Recurring
130     .RecField1.Value = RecID
131     .RecField2.Value = TransType
132     .RecField3.Value = TransID
133     .RecField4.Value = CustNm
134     .RecField5.Value = RecurDB.Range("E" & RecRow).Value 'Recurring Frequency
135     .RecField6.Value = RecurDB.Range("F" & RecRow).Value 'Recurring
136     .RecField7.Value = RecurDB.Range("G" & RecRow).Value 'Next Date
137     .Show
138 End With
139
140 End With
141 End Sub
142 Sub Recurring_SaveUpdate()
143     With Recurring
144         'Check For Required Fields
145         If .RecField5.Value = "" Or .RecField6.Value = Empty Then 'Check For Rec. # and
146             MsgBox "Please make sure to add in a Frequency Qty and Frequency before
147                 saving"
148             Exit Sub
149         End If
150         If IsNumeric(.RecField5.Value) = False Then
151             MsgBox "Please make sure the Frequency Qty is a whole #"
152             Exit Sub
153         End If
154         RecID = .RecField1.Value 'Recurring Id
155         On Error Resume Next
156         RecRow = RecurDB.Range("Rec_ID").Find(RecID, , xlValues, xlWhole).Row
157         On Error GoTo 0
158         If RecRow = 0 Then RecRow = RecurDB.Range("A9999").End(xlUp).Row + 1 'First
159             AvailRow
160         For RecCol = 1 To 6
161             Set RecurFld = .Controls("RecField" & RecCol)
162             RecurDB.Cells(RecRow, RecCol).Value = RecurFld.Value 'Save to Recurring
163             database
164         Next RecCol
165         RecurDB.Range("G" & RecRow).Value = .RecField7.Value
166         ActiveSheet.Range("B7").Value = .RecField1.Value 'Set Recurring ID
167         .Hide
168         If ActiveSheet.Name = "Invoice" Then Invoice_SaveUpdate Else Payment_SaveUpdate
169         'Save Invoice or Payment
170     End With
171 End Sub

```

_, 6, 12, 34

A

Activate, 21, 32
 ActiveCell, 31, 36
 ActiveSheet, 30-32, 34-37, 39, 40
 ActiveWorkbook, 31-33
 ActSht, 32
 Add, 7, 24, 27, 32, 33
 AddCustForm, 6, 19, 24
 Address, 31, 32
 Admin, 6, 25, 29
 AdvancedFilter, 21, 22, 24, 25, 27, 38
 After, 32
 Aging_GoToInvoice, 21
 Aging_Refresh, 21, 22
 Aging_ShowCustDetail, 5, 21
 AgingDetail_Refresh, 21, 22
 Application, 19, 22, 23, 30, 36, 40
 Apply, 24, 27
 Array, 30, 31, 34-36
 AutoFill, 33

B

Bold, 30, 31, 35

C

Cal, 7, 8, 12
 CalCol, 30, 32
 CalendarHide, 8, 12, 30
 CalendarShow, 8, 12, 30
 CalFormulaReplacement, 31
 Caller, 22, 30, 36
 CalPopUp, 36
 CancelBtn_Click, 19, 20
 Cells, 19, 31, 32, 36, 40
 CheckForSheet, 8, 12, 31
 Chr, 3, 25, 32, 33
 Clear, 24, 27
 ClearContents, 3, 5-7, 21, 22, 24, 25, 27, 38, 39
 ClientNmLbl_Click, 20
 ColCnt, 32
 ColNum, 31
 ColNumb, 36
 Column, 5
 Control, 19, 38
 Controls, 19, 40
 Copy, 32, 36
 CopyToRange, 21, 22, 24, 25, 27, 38
 Count, 32
 CountLarge, 3, 5-7, 12
 CreateCalSht, 32
 CreateWS, 31, 32
 CriteriaRange, 21, 22, 24, 25, 27, 38
 CustCol, 19
 CustNm, 38-40
 Customer_AddNew, 24
 CustomerFld, 19
 Customers, 6, 12, 19
 CustRow, 6, 12, 19

D

Dashboard, 21, 22
 Dashboard_Refresh, 22
 Dashboard_SelectAgingDetails, 22
 Dashboard_TabChange, 5, 22
 DataOption, 24, 27

DateAdd, 38, 39
 DayCnt, 32
 DayName, 30, 31, 35
 DayNum, 30, 31
 DayNumb, 36
 Delay, 25, 26, 28
 Delete, 7, 24, 27, 39
 DeleteBtn_Click, 20
 Destination, 33
 DetailNumb, 21, 22
 Dir, 25
 DoEvents, 26, 28
 DrawingObject, 31, 32

E

Empty, 3, 5-7, 12, 19, 24, 26, 27, 29, 30, 35, 36, 39, 40
 EntireRow, 7, 22, 24, 27, 39
 Explicit, 3-6, 9-21, 24, 27, 38
 ExportAsFixedFormat, 25

F

Field1, 6, 19
 FilePath, 25
 Fill, 26, 28, 30, 31, 35
 Find, 6, 7, 25, 38-40
 Font, 30, 31, 35
 ForeColor, 30, 31, 35
 Formula, 21, 22, 31, 32
 FreqType, 38, 39

G

Group, 34
 GroupCal, 33, 34
 GroupItems, 31

H

Hidden, 22
 Hide, 19, 20, 39, 40
 HorizontalAlignment, 21

I

i, 25, 26, 28
 InStr, 31, 35
 Intersect, 3, 5-8, 12
 InvItemDBRow, 7
 InvItemRow, 24-26
 Invoice, 19, 21, 24-26, 38, 39
 Invoice_Delete, 24
 Invoice_Load, 6, 24, 38
 Invoice_New, 24, 25
 Invoice_Print, 25
 Invoice_SaveAsPDF, 25
 Invoice_SavedMsg, 25, 26
 Invoice_SaveUpdate, 25, 26, 38, 40
 Invoiceltems, 7, 24-26
 Invoicelist, 21, 22, 24-26, 29
 InvRow, 24-27, 29
 IsDate, 30
 IsNumeric, 5, 39, 40
 ItemRow, 6, 7, 24, 26
 Items, 6, 7

K

Key, 24, 27
 Kill, 25

L

LastItemRow, 24, 26
 LastResultRow, 21, 22, 24, 25, 27, 38

LastRow, 21, 22, 24, 25, 27, 38
 Left, 5, 31, 35
 Len, 35

M

MacroLink, 34, 35
 MacroLinkRemover, 31, 34
 Max, 19, 40
 Month, 30, 32
 MsgBox, 6, 19, 24, 26-29, 31, 34, 39, 40
 msoCTrue, 5, 22, 25, 28, 31, 34
 msoFalse, 5, 22, 26, 28, 30, 35, 36
 msoTrue, 31

N

Name, 32-34, 39, 40
 Names, 33
 NewLink, 34, 35
 NextMonth, 35
 NextShp, 34, 35
 NextTransDt, 38, 39
 NextYear, 35
 NoCal, 30, 31, 33
 Noltems, 25, 26
 NotSaved, 24, 27, 39
 NumberFormat, 33

O

Offset, 31, 36
 OnAction, 31, 32, 35
 Order, 24, 27

P

Paste, 32, 36
 Path, 25
 Payment_CustomerPmntsRefresh, 12, 27, 29
 Payment_Delete, 27
 Payment_Load, 27, 28, 38
 Payment_New, 27
 Payment_Next, 28
 Payment_Prev, 28
 Payment_SavedMsg, 28, 29
 Payment_SaveUpdate, 29, 38, 40
 Payments, 27-29, 38, 39
 Placement, 31, 34
 PmntID, 27, 28
 PmntRow, 27, 29
 PmntsDB, 27, 29
 PmtRow, 6
 PmtTerm, 6
 PrevMonth, 35
 PrevYear, 36
 PrintOut, 25

R

Range, 3, 5-8, 12, 19, 21, 22, 24-36, 38-40
 Rec_ID, 40
 RecCol, 38, 40
 RecField1, 39, 40
 RecField2, 40
 RecField3, 40
 RecField4, 40
 RecField5, 39, 40
 RecField5_Change, 20
 RecField6, 39, 40
 RecField6_AfterUpdate, 20
 RecField6_Change, 20
 RecField7, 39, 40
 RecFreq, 38, 39
 RecID, 38-40

- RecRow, [38-40](#)
 RecurDB, [38-40](#)
 RecurFld, [38, 40](#)
 Recurring, [39, 40](#)
 Recurring_CheckAndCreate, [38](#)
 Recurring_Delete, [20, 39](#)
 Recurring_FrequencyChange, [20, 39](#)
 Recurring_Open, [39](#)
 Recurring_SaveUpdate, [20, 40](#)
 RefersTo, [33](#)
 Replace, [22, 36](#)
 ReplaceCalendar, [8, 12, 36](#)
 ResultRow, [24, 25, 38](#)
 RGB, [30, 31, 35](#)
 Right, [35](#)
 RoundUp, [36](#)
 Row, [3, 5-7, 19, 21, 22, 24-27, 29, 38-40](#)
 RowCnt, [32](#)
 RowNum, [31](#)
 RowNumb, [36](#)
- S**
- SaveBtn_Click, [19, 20](#)
 ScreenUpdating, [22, 23](#)
 SelCell, [30, 31](#)
 SelCol, [21, 22](#)
 SelectDay, [36](#)
 Selection, [30, 34](#)
 SelRow, [21](#)
 SetRange, [24, 27](#)
 Shape, [7, 12, 31, 34](#)
 ShapeRange, [30, 34](#)
 Shapes, [5, 7, 12, 22, 25, 28, 30-32, 34-37](#)
 Sheets, [30-32, 35, 36](#)
 Show, [6, 24, 40](#)
 ShowSettings, [36](#)
 Shp, [31, 34, 35](#)
 SingleRow, [24](#)
 SkipSort, [27](#)
 Sort, [24, 27](#)
 SortFields, [24, 27](#)
 SortOn, [24, 27](#)
 Split, [35](#)
 SplitLink, [34, 35](#)
 StartTime, [25, 26, 28](#)
 StatusRow, [24, 25](#)
- T**
- Target, [3, 5-8, 12](#)
 TechRow, [7](#)
 Techs, [7](#)
 TermRow, [24, 25](#)
 TextFrame2, [30, 31, 35](#)
 TextRange, [30, 31, 35](#)
 ThisWorkbook, [25, 32, 35, 36](#)
 Timer, [26, 28](#)
 Top, [5, 31](#)
 TransDt, [38](#)
 TransID, [38-40](#)
 Transparency, [26, 28](#)
 TransType, [38-40](#)
- U**
- Ungroup, [36, 37](#)
 UnGroupCal, [32, 36](#)
 Unique, [21, 22, 24, 25, 27, 38](#)
 Unload, [24](#)
- V**
- Validation, [7](#)
 Value, [3, 5-7, 12, 19, 21, 22, 24-33, 35, 36, 38-40](#)
 vbDirectory, [25](#)
 vbNo, [24, 27, 29, 39](#)
 vbYes, [6](#)
 vbYesNo, [6, 24, 27, 29, 39](#)
 Visible, [5, 22, 25, 26, 28, 30-32, 34-36](#)
- W**
- Worksheet, [31, 32](#)
 Worksheet_Change, [6, 12](#)
 Worksheet_SelectionChange, [3, 5, 7, 12](#)
 WorksheetFunction, [19, 36, 40](#)
 ws, [31, 32](#)
- X**
- xlBetween, [7](#)
 xlCenterAcrossSelection, [21](#)
 xlDescending, [24, 27](#)
 xlFillDefault, [33](#)
 xlFilterCopy, [21, 22, 24, 25, 27, 38](#)
 xlMove, [31, 34](#)
 xlSheetHidden, [32](#)
 xlSortNormal, [24, 27](#)
 xlSortOnValues, [24, 27](#)
 xlTypePDF, [25](#)
 xlUp, [19, 21, 22, 24-27, 29, 38, 40](#)
 xlValidAlertStop, [7](#)
 xlValidateList, [7](#)
 xlValues, [6, 7, 25, 38-40](#)
 xlWhole, [6, 7, 25, 38-40](#)
- Y**
- Year, [30, 32](#)

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