

VBA SOURCE CODE BOOK

ACCOUNTS RECEIVABLE MANAGER

ACCOUNTS RECEIVABLE DASHBOARD

TOTAL PAID VS. OUTSTANDING BALANCES

Total Overdue	\$1,222.00
Total Due	\$1,437.50

AGING SUMMARY

Category	Amount
Current Bal.	\$413.50
1-30 Days	\$900.50
31-60 Days	\$419.00
61-90 Days	\$4.00
Older Bal.	\$900.50

TOP 5 CUSTOMERS BY SALES INVOICED

Customer	Amount
Customer 1	\$764.50
Customer 2	\$550.00
Customer 3	\$585.00
Customer 4	\$286.00
Customer 5	\$88.00

TOP 5 CUSTOMERS BY AMOUNT DUE

Customer	Amount
Customer 1	\$529.00
Customer 2	\$350.00
Customer 3	\$166.00
Customer 4	\$166.00
Customer 5	\$166.00

SERVICE INVOICE Invoice #: 4

Fredders Garage
129 Main St.
Anytown, USA 90055
310 555-1212

Customer: Frank Costanza
12345 Main St.
Los Angeles CA, 90232

Invoice Date: 4/5/2022
Status: Partial Paid
Terms: Net 15
Due On: 4/20/2022

DATE	TECH./ITEM	DESCRIPTION	QTY	AMOUNT	TOTAL
4/6/2022	22cm Brake Pads	Fredders 22cm Brake Pads	1.00	\$25.00	\$ 25.00
	Fredders	Standard Break Repair	1.00	\$100.00	\$ 100.00

ENTER PAYMENTS

Enter Invoice # 5
Payment Date: 4/7/2022

Customer: Dolores Richman
438-5218 Suspensidse Rd.
Racine Wisconsin, 68816

Invoice Amount: \$264
Previous Payments: \$262
Payment Amount: \$2.00
Invoice Balance: \$2.00

Notes: 2. Payment

Customer Payment History

Payment Date	Amount	Inv. #	Notes
4/7/2022	\$190.00	5	
4/7/2022	\$70.00	5	
4/7/2022	\$2.00	5	2. Payment



Excel For Freelancers

Create Your Own Accounts Receivable Manager 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



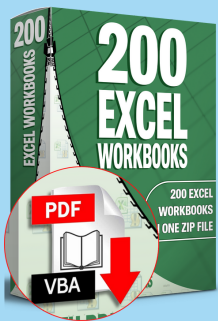
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



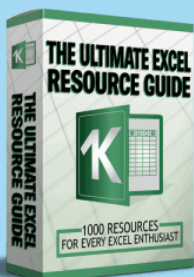
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[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](#)

Table of Contents

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
Sheet1.....	14
(Declarations).....	14
Sheet6.....	15
(Declarations).....	15
Techs.....	16
(Declarations).....	16
ThisWorkbook.....	17
(Declarations).....	17
Forms.....	18
AddCustForm.....	18
(Declarations).....	18
CancelBtn_Click [Sub].....	18
SaveBtn_Click [Sub].....	18
Modules.....	19
Dashboard_Macros.....	19
(Declarations).....	19
Aging_GoToInvoice [Sub].....	19
Aging_Refresh [Sub].....	19
Aging_ShowCustDetail [Sub].....	19
AgingDetail_Refresh [Sub].....	19
Dashboard_Refresh [Sub].....	20
Dashboard_SelectAgingDetails [Sub].....	20
Dashboard_TabChange [Sub].....	20
Invoice_Macros.....	22
(Declarations).....	22
Customer_AddNew [Sub].....	22
Invoice_Delete [Sub].....	22
Invoice_Load [Sub].....	22
Invoice_New [Sub].....	23
Invoice_Print [Sub].....	23
Invoice_SaveAsPDF [Sub].....	23
Invoice_SavedMsg [Sub].....	23
Invoice_SaveUpdate [Sub].....	24
Pmnt_Macros.....	25
(Declarations).....	25
Payment_CustomerPmntsRefresh [Sub].....	25
Payment_Delete [Sub].....	25
Payment_Load [Sub].....	25
Payment_New [Sub].....	25

Table of Contents

Payment_Next [Sub]	25
Payment_Prev [Sub]	26
Payment_SavedMsg [Sub]	26
Payment_SaveUpdate [Sub]	26
PopUp_Calendar	28
(Declarations)	28
CalCol [Sub]	28
CalendarHide [Sub]	28
CalendarShow [Sub]	28
CalFormulaReplacement [Sub]	29
CheckForSheet [Sub]	29
CreateCalSht [Sub]	30
GroupCal [Sub]	31
MacroLinkRemover [Sub]	32
NextMonth [Sub]	33
NextYear [Sub]	33
PrevMonth [Sub]	33
PrevYear [Sub]	34
ReplaceCalendar [Sub]	34
SelectDay [Sub]	34
ShowSettings [Sub]	34
UnGroupCal [Sub]	34

```
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
19     If Not Intersect(Target, Range("J37:R99999" )) Is Nothing And Range("J" & Target.
20         Row).Value <> Empty And IsNumeric(Range("J" & Target.Row).Value) = True Then
21         Range("AA2" ).Value = Target.Row
22         With Shapes("ViewInvBtn" )
23             .Left = Range("S" & Target.Row).Left
24             .Top = Range("S" & Target.Row).Top
25             .Visible = msoCTrue
26         End With
27     Else
28         Range("AA2" ).ClearContents
29     End If
30 End Sub
```



```
1 Option Explicit
```

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

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

```
4     'On customer name change
```

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

```
6     Then
```

```
7         Dim CustRow As Long
```

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

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

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

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

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

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

```
14        Else 'Customer Not Found
```

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

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

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

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

```
19                AddCustForm.Show
```

```
20            End If
```

```
21        End If
```

```
22    End If
```

```
23
```

```
24
```

```
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
```

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

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

```
39     Then
```

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

```
41         Dim PmtTerm As String
```

```
42         Dim PmtRow As Long
```

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

```
44         On Error Resume Next
```

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

```
46         On Error GoTo 0
```

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

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

```
49     End If
```

```
50
```

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

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

```
53     Then
```

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

```
55             Dim ItemRow As Long
```

```
56             On Error Resume Next
```

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

```
58             'Item Row (if Found)
```

```
1 2 3
```

```

1 2 3
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" )
101    On Error GoTo 0
102    If Cal Is Nothing Then ReplaceCalendar 'Run Macro to Replace Calendar

```

```
1
103
104 If Not Intersect(Target, Range("J3,E9:E31" )) Is Nothing Then 'Check For Sheet and
    Calendard
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
2

Option Explicit

1
2

Option Explicit

1
2

Option Explicit

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 Dim LastRow As Long, LastResultRow As Long, SelRow As Long, SelCol As Long, DetailNum
  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
        Paid & Days Overdue Formulas
22         .Range("A2:D" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
        "L1:L2" ), CopyToRange:=.Range("P2" ), Unique:=True
23         LastResultRow = .Range("P99999" ).End(xlUp).Row
24         If LastResultRow < 3 Then Exit Sub
25         .Range("Q3:V" & LastResultRow).Formula = .Range("Q1:W1" ).Formula
26         Dashboard.Range("B35:H" & LastResultRow + 32).Value = .Range("P3:V" &
        LastResultRow).Value 'Bring over Aging Data
27     End With
28 End Sub
29
30
31 Sub Aging_ShowCustDetail()
32     Dashboard.Range("J34:R499" ).ClearContents 'Clear Previous Results
33     SelRow = Dashboard.Range("AA1" ).Value 'Set Selected Row
34     With Invoicelist
35         'Clear Prior Results
36         .Range("AB3:AJ9999" ).ClearContents
37         LastRow = .Range("A99999" ).End(xlUp).Row
38         If LastRow < 3 Then Exit Sub
39         .Range("H3:J" & LastRow).Formula = .Range("H1:J1" ).Formula 'Bring Down Total
        Paid & Days Overdue Formulas
40         .Range("A2:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
        "Y1:Z2" ), CopyToRange:=.Range("AB2:AJ2" ), Unique:=True
41         LastResultRow = .Range("AB99999" ).End(xlUp).Row
42         If LastResultRow < 3 Then Exit Sub
43         Dashboard.Range("J" & SelRow & ":R" & SelRow + LastResultRow - 1).Value = .
        Range("AB1:AJ" & LastResultRow).Value 'Bring over Customer Details
44         Dashboard.Range("J" & SelRow & ":R" & SelRow).HorizontalAlignment =
        xlCenterAcrossSelection
45     End With
46 End Sub
47
48
49 Sub AgingDetail_Refresh()
50     Dashboard.Range("B507:J9999" ).ClearContents 'Clear Existing Data
51     With Invoicelist
52         'Clear Prior Results
53         .Range("AB3:AJ9999" ).ClearContents

```

```

1 2
54 LastRow = .Range("A99999" ).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(
    "A1: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("A99999" ).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     End With
109     Application.ScreenUpdating = True

```

110

1

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("A99999" ).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("P99999" ).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 enter 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("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
56

```

```

1 2
57 With InvoiceItems
58     LastRow = .Range("A99999").End(xlUp).Row
59     If LastRow < 3 Then GoTo NoItems
60     .Range("A2:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
        "M2:M3" ), CopyToRange:=.Range("P2:Y2" ), Unique:=True
61     LastResultRow = .Range("P99999").End(xlUp).Row
62     If LastResultRow < 3 Then GoTo NoItems
63     For ResultRow = 3 To LastResultRow
64         InvItemRow = .Range("Y" & ResultRow).Value 'Get Invoice Row
65         Invoice.Range("B" & InvItemRow & ":I" & InvItemRow).Value = .Range("P" &
            ResultRow & ":W" & ResultRow).Value 'Item Details
66         Invoice.Range("K" & InvItemRow).Value = InvoiceItems.Range("X" &
            ResultRow).Value 'Item Cost
67     Next ResultRow
68 End With
69 NoItems:
70     .Range("B6").Value = False 'Set inv. Load to false
71 End With
72 End Sub

73
74 Sub Invoice_New()
75     With Invoice
76         .Range("B6").Value = True 'Set Inv. Load to true
77         .Range("I3:J6,G5:G7,B9:I31,K9:K31").ClearContents
78         .Range("J1").Value = .Range("B5").Value 'Set Next invoice #
79         .Range("I3").Value = Date 'Set current Date
80         .Range("B6").Value = False 'Set inv. Load to false
81     On Error Resume Next
82     TermRow = Admin.Range("H6:H23").Find(Chr(252), , xlValues, xlWhole).Row
83     On Error GoTo 0
84     If TermRow <> 0 Then .Range("I5").Value = Admin.Range("F" & TermRow).Value
        'Set Default Term
85     On Error Resume Next
86     StatusRow = Admin.Range("D6:D12").Find(Chr(252), , xlValues, xlWhole).Row
87     On Error GoTo 0
88     If StatusRow <> 0 Then .Range("I4").Value = Admin.Range("C" & StatusRow).Value
        'Set Default Status
89 End With
90 End Sub

91
92
93 Sub Invoice_Print()
94     Invoice.PrintOut , , , True, , , False
95 End Sub

96
97 Sub Invoice_SaveAsPDF()
98     Dim FilePath As String
99     Invoice_SaveUpdate 'Save invoice
100    FilePath = ThisWorkbook.Path & "\" & Invoice.Range("G5").Value & "_" & Invoice.
        Range("J1").Value 'File Path
101    If Dir(FilePath, vbDirectory) <> "" Then Kill (FilePath)
102    Invoice.ExportAsFixedFormat xlTypePDF, FilePath, , , False, , , True
103 End Sub

104 Sub Invoice_SavedMsg()
105     With Invoice.Shapes("InvSavedMsg" )
106         Dim i As Long, Delay As Double, StartTime As Double
107         .Visible = msoCTrue
108         For i = 1 To 150
109             .Fill.Transparency = i / 150
110             Delay = 0.009
111             StartTime = Timer

```

```

112 1 2 3
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         'Deterermine New Invoice/Existing Invoice
127         If .Range("B3").Value = Empty Then 'new Invoice
128             InvRow = InvoiceList.Range("A99999").End(xlUp).Row + 1 'First Avail Row
129             .Range("J1").Value = .Range("B5").Value 'Next Inv. #
130             InvoiceList.Range("A" & InvRow).Value = .Range("B5").Value 'Next Inv. #
131         Else 'Existing Invoice
132             InvRow = .Range("B3").Value 'Invoice Row
133         End If
134         InvoiceList.Range("B" & InvRow).Value = .Range("I3").Value 'Date
135         InvoiceList.Range("C" & InvRow).Value = .Range("G5").Value 'Customer
136         InvoiceList.Range("D" & InvRow).Value = .Range("I4").Value 'Status
137         InvoiceList.Range("E" & InvRow).Value = .Range("I5").Value 'Terms
138         InvoiceList.Range("F" & InvRow).Value = .Range("I6").Value 'Due Date
139         InvoiceList.Range("G" & InvRow).Value = .Range("J34").Value 'Invoice Total
140
141
142         'Add/Update Invoice Items
143         LastItemRow = .Range("C31").End(xlUp).Row 'Last Invoice Row
144         If LastItemRow < 9 Then GoTo NoItems
145         For ItemRow = 9 To LastItemRow
146             If .Range("B" & ItemRow).Value <> Empty Then 'DB Row Exists
147                 InvItemRow = .Range("B" & ItemRow).Value 'Inv. Item DB Row
148             Else 'New DB ItemRow
149                 InvItemRow = InvoiceItems.Range("A99999").End(xlUp).Row + 1 'First avail
150                 InvoiceItems.Range("A" & InvItemRow).Value = .Range("J1").Value 'Inv ID
151                 .Range("B" & ItemRow).Value = InvItemRow 'Add DB Row
152                 InvoiceItems.Range("K" & InvItemRow).Value = "=Row()" 'Add Row Formula
153             End If
154             InvoiceItems.Range("B" & InvItemRow & ":H" & InvItemRow).Value = .Range("C"
155                 & ItemRow & ":I" & ItemRow).Value ' Bring over item Values
156             InvoiceItems.Range("I" & InvItemRow).Value = .Range("K" & ItemRow).Value
157             'Item Cost
158             InvoiceItems.Range("J" & InvItemRow).Value = ItemRow 'Invoice Row
159         Next ItemRow
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("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         If .Range("B6").Value <> "" Then Payment_CustomerPmntsRefresh 'Load Previous Customer Payments
49     End With
50 End Sub
51 Sub Payment_New()
52     Payments.Range("B2,E3,H7,D5:D7,E9:H9,D13:H999").ClearContents
53     Payments.Range("E3").Select
54 End Sub
55

```

```

56 Sub Payment_Next()
57     With Payments
58         PmntID = .Range("B2").Value 'Payment ID
59         If PmntID = 0 Then 'No Current ID
60             If .Range("B4").Value = 1 Then 'No Saved Payments
61                 MsgBox "Please save any Payments first before navigating to previously
62                     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
84                     saved"
85                 Exit Sub
86             End If
87             .Range("B2").Value = .Range("B4").Value - 1 'Set Pmnt. ID to the last one
88             created
89             Payment_Load
90             Exit Sub
91         End If
92         If PmntID = 1 Then
93             MsgBox "You are already at the first Payment created"
94             Exit Sub
95         End If
96         .Range("B2").Value = .Range("B2").Value - 1 'Set Previous Payment ID
97         Payment_Load
98     End With
99 End Sub
100
101 Sub Payment_SavedMsg()
102     With Payments.Shapes("PmntSavedMsg")
103         Dim i As Long, Delay As Double, StartTime As Double
104         .Visible = msoCTrue
105         For i = 1 To 150
106             .Fill.Transparency = i / 150
107             Delay = 0.009
108             StartTime = Timer
109             Do
110                 DoEvents
111             Loop While Timer - StartTime < Delay
112         Next i
113         .Visible = msoFalse
114     End With
115 End Sub
116
117 Sub Payment_SaveUpdate()
118     With Payments

```

```

1 2
114 If .Range("B5").Value = Empty Then 'Incorrect / Missing Invoice #
115     MsgBox "Please select a correct Invoice #"
116     Exit Sub
117 End If
118 InvRow = .Range("B5").Value 'Invoice Row
119 If .Range("H3").Value = Empty Or .Range("H7").Value = Empty Then 'Empty Fields
120     MsgBox "Please make sure to add in a Payment Date and Payment Amount"
121     Exit Sub
122 End If
123
124 If .Range("H8").Value < 0 Then
125     If MsgBox("The Payment Amount is above the Invoice Balance. Are you sure you
126         want to continue?", vbYesNo, "Payment Amount Issue" ) = vbNo Then Exit Sub
127 End If
128
129 If .Range("B3").Value = Empty Then 'New Payment
130     PmntRow = PmntsDB.Range("A99999").End(xlUp).Row + 1
131     .Range("B2").Value = .Range("B4").Value 'Next Pment ID
132     PmntsDB.Range("A" & PmntRow).Value = .Range("B4").Value 'Next Payment ID
133 Else 'Existing Payment
134     PmntRow = .Range("B3").Value 'Payment Row
135 End If
136 PmntsDB.Range("B" & PmntRow).Value = .Range("H3").Value 'Date
137 PmntsDB.Range("C" & PmntRow).Value = .Range("D5").Value 'Customer
138 PmntsDB.Range("D" & PmntRow).Value = .Range("E3").Value 'Invoice #
139 PmntsDB.Range("E" & PmntRow).Value = .Range("H7").Value 'Amount
140 PmntsDB.Range("F" & PmntRow).Value = .Range("E9").Value 'Notes
141
142 'Update invoice Paid Status
143 If .Range("H8").Value = 0 Then 'Paid
144     InvoiceList.Range("D" & InvRow).Value = Admin.Range("C10").Value 'Fully
145     Paid
146 Else ' Partial Paid
147     InvoiceList.Range("D" & InvRow).Value = Admin.Range("C9").Value 'Partially
148     Paid
149 End If
150 End With
151 Payment_CustomerPmntsRefresh 'Refresh Customer Payments
152 Payment_SavedMsg 'Run Fade out message
153 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
58         If InStr(.Shapes("Calendar" ).GroupItems("NextTri" ).OnAction, "!" ) <> 0 Or

```

1 2 3


```

1 2 3
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
111     CreateWS:
112     CreateCalSht

```



```

1
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"
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 2

```

1 2
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$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
176 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
    ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($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:
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()

```

```

1
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
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)

```

1 2 3

```

255     1 2 3 SplitLink = Split(MacroLink, "!" )
256
257     'Pull text occurring after exclamation mark
258     NewLink = SplitLink(1)
259
260     'Remove any stragglng 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
309     With Sheets("CalPopUp" )
310         If .Range("A3" ).Value = 1 Then
311             .Range("A3" ).Value = 12
312             .Range("A2" ).Value = .Range("A2" ).Value - 1
313         Else:
314             .Range("A3" ).Value = .Range("A3" ).Value - 1
315         End If
316     End With
317 End Sub

```

```

1 2 3
313     End If
314     If .Range("A6" ).Value > 0 Then
315         ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
            "40Day" , "41Day" , "42Day" )).Visible = True
316     Else:
317         ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
            "40Day" , "41Day" , "42Day" )).Visible = False
318     End If
319 End With
320 End Sub

321
322 Sub PrevYear()
323     ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
        "CalPopUp" ).Range("A2" ).Value - 1
324 End Sub

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

330
331 .....
332 ''''Select Day Of The Month
333 .....

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

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

354
355 Sub UnGroupCal()
356     On Error Resume Next
357     ActiveSheet.Shapes("Calendar" ).Ungroup
358     ActiveSheet.Shapes("NextMonth" ).Ungroup
359     ActiveSheet.Shapes("PrevMonth" ).Ungroup
360     On Error GoTo 0
361 End Sub

```

—, 6, 12, 32

A

Activate, 19, 30
 ActiveCell, 29, 34
 ActiveSheet, 28-30, 32-34
 ActiveWorkbook, 29-31
 ActSht, 30
 Add, 7, 22, 25, 30, 31
 AddCustForm, 6, 18, 22
 Address, 29, 30
 Admin, 6, 23, 27
 AdvancedFilter, 19, 20, 22, 23, 25
 After, 30
 Aging_GoToInvoice, 19
 Aging_Refresh, 19, 20
 Aging_ShowCustDetail, 5, 19
 AgingDetail_Refresh, 19, 20
 Application, 18, 20, 28, 34
 Apply, 22, 25
 Array, 28, 29, 32-34
 AutoFill, 30, 31

B

Bold, 28, 29, 33

C

Cal, 7, 12
 CalCol, 28, 30
 CalendarHide, 8, 12, 28
 CalendarShow, 8, 12, 28
 CalFormulaReplacement, 29
 Caller, 20, 28, 34
 CalPopUp, 34
 CancelBtn_Click, 18
 Cells, 18, 29, 30, 34
 CheckForSheet, 8, 12, 29
 Chr, 3, 23, 30, 31
 Clear, 22, 25
 ClearContents, 3, 5-7, 19, 20, 22, 23, 25
 ColCnt, 30
 ColNum, 29
 ColNumb, 34
 Column, 5
 Control, 18
 Controls, 18
 Copy, 30, 34
 CopyToRange, 19, 20, 22, 23, 25
 Count, 30
 CountLarge, 3, 5-7, 12
 CreateCalSht, 29, 30
 CreateWS, 29
 CriteriaRange, 19, 20, 22, 23, 25
 CustCol, 18
 Customer_AddNew, 22
 CustomerFld, 18
 Customers, 6, 12, 18
 CustRow, 6, 12, 18

D

Dashboard, 19, 20
 Dashboard_Refresh, 20
 Dashboard_SelectAgingDetails, 20
 Dashboard_TabChange, 5, 20
 DataOption, 22, 25
 DayCnt, 30
 DayName, 28, 29, 33
 DayNum, 28, 29

DayNumb, 34
 Delay, 23, 24, 26
 Delete, 7, 22, 25
 Destination, 30, 31
 DetailNumb, 19, 20
 Dir, 23
 DoEvents, 24, 26
 DrawingObject, 29, 30

E

Empty, 3, 5-7, 12, 18, 22, 24, 25, 27, 28, 33, 34
 EntireRow, 7, 20, 22, 25
 Explicit, 3-6, 9-19, 22, 25
 ExportAsFixedFormat, 23

F

Field1, 6, 18
 FilePath, 23
 Fill, 23, 26, 28, 29, 33
 Find, 6, 7, 23
 Font, 28, 29, 33
 ForeColor, 28, 29, 33
 Formula, 19, 20, 29, 30

G

Group, 32
 GroupCal, 31
 GroupItems, 28

H

Hidden, 20
 Hide, 18
 HorizontalAlignment, 19

I

i, 23, 24, 26
 InStr, 28, 29, 32
 Intersect, 3, 5-8, 12
 InvItemDBRow, 7
 InvItemRow, 22-24
 Invoice, 18, 19, 22-24
 Invoice_Delete, 22
 Invoice_Load, 6, 22
 Invoice_New, 22, 23
 Invoice_Print, 23
 Invoice_SaveAsPDF, 23
 Invoice_SavedMsg, 23, 24
 Invoice_SaveUpdate, 23, 24
 InvoicItems, 7, 22-24
 Invoicelist, 19, 20, 22, 24, 27
 InvRow, 22, 24, 25, 27
 IsDate, 28
 IsNumeric, 5
 ItemRow, 6, 7, 22, 24
 Items, 6, 7

K

Key, 22, 25
 Kill, 23

L

LastItemRow, 22, 24
 LastResultRow, 19, 20, 22, 23, 25
 LastRow, 19, 20, 22, 23, 25
 Left, 5, 29, 33
 Len, 33

M

MacroLink, 32, 33

MacroLinkRemover, 29, 32
 Max, 18
 Month, 28, 30
 MsgBox, 6, 18, 22, 24-27, 29, 31
 msoCTrue, 5, 20, 23, 26, 29, 32
 msoFalse, 5, 20, 24, 26, 28, 33, 34
 msoTrue, 29

N

Name, 30-32
 Names, 31
 NewLink, 32, 33
 NextMonth, 33
 NextShp, 32, 33
 NextYear, 33
 NoCal, 28, 29, 31
 Noltems, 23, 24
 NotSaved, 22, 25
 NumberFormat, 31

O

Offset, 29, 34
 OnAction, 28, 30, 32, 33
 Order, 22, 25

P

Paste, 30, 34
 Path, 23
 Payment_CustomerPmntsRefresh, 12, 25, 27
 Payment_Delete, 25, 26
 Payment_Load, 25, 26
 Payment_New, 25
 Payment_Next, 26
 Payment_Prev, 26
 Payment_SavedMsg, 26, 27
 Payment_SaveUpdate, 26
 Payments, 25, 26
 Placement, 29, 32
 PmntID, 25, 26
 PmntRow, 25, 27
 PmntsDB, 25, 27
 PmtRow, 6
 PmtTerm, 6
 PrevMonth, 33
 PrevYear, 34
 PrintOut, 23

R

Range, 3, 5-8, 12, 18-20, 22-34
 RefersTo, 31
 Replace, 20, 34
 ReplaceCalendar, 7, 12, 34
 ResultRow, 22, 23
 RGB, 28, 29, 33
 Right, 33
 RoundUp, 34
 Row, 3, 5-7, 18-20, 22-25, 27
 RowCnt, 30
 RowNum, 29
 RowNumb, 34

S

SaveBtn_Click, 18
 ScreenUpdating, 20
 SelCell, 28, 29
 SelCol, 19, 20
 SelectDay, 34
 Selection, 28, 32
 SelRow, 19
 SetRange, 22, 25

Shape, 7, 12, 29, 32
ShapeRange, 28, 32
Shapes, 5, 7, 12, 20, 23, 26, 28-30, 32-34
Sheets, 28-30, 33, 34
Show, 6, 22
ShowSettings, 34
Shp, 29, 32, 33
SingleRow, 22
SkipSort, 25
Sort, 22, 25
SortFields, 22, 25
SortOn, 22, 25
Split, 33
SplitLink, 32, 33
StartTime, 23, 24, 26
StatusRow, 22, 23

T

Target, 3, 5-8, 12
TechRow, 7
Techs, 7
TermRow, 22, 23
TextFrame2, 28, 29, 33
TextRange, 28, 29, 33
ThisWorkbook, 23, 30, 33, 34
Timer, 23, 24, 26
Top, 5, 29
Transparency, 23, 26

U

Ungroup, 34
UnGroupCal, 30, 34
Unique, 19, 20, 22, 23, 25
Unload, 22

V

Validation, 7
Value, 3, 5-7, 12, 18-20, 22-31, 33, 34
vbDirectory, 23
vbNo, 22, 25, 27
vbYes, 6
vbYesNo, 6, 22, 25, 27
Visible, 5, 20, 23, 24, 26, 28-30, 32-34

W

Worksheet, 29, 30
Worksheet_Change, 6, 12
Worksheet_SelectionChange, 3, 5, 7, 12
WorksheetFunction, 18, 34
ws, 29, 30

X

xlBetween, 7
xlCenterAcrossSelection, 19
xlDescending, 22, 25
xlFillDefault, 30, 31
xlFilterCopy, 19, 20, 22, 23, 25
xlMove, 29, 32
xlSheetHidden, 30
xlSortNormal, 22, 25
xlSortOnValues, 22, 25
xlTypePDF, 23
xlUp, 18-20, 22-25, 27
xlValidAlertStop, 7
xlValidateList, 7
xlValues, 6, 7, 23
xlWhole, 6, 7, 23

Y

Year, 28, 30

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