

VBA SOURCE CODE BOOK

INVOICE INSTALLMENT PAYMENTS

EXCLUSIVE FOR SILVER & GOLD MEMBERS

UPDATED

Due Date	Print Amnt.	Paid On	Paid Amnt.	Balance
12/1/2023	\$127.75	12/1/2023	\$127.75	\$1,400.25
1/1/2024	\$127.75	1/5/2024	\$127.75	\$1,372.50
2/1/2024	\$127.75	2/1/2024	\$127.75	\$1,244.75
3/1/2024	\$127.75	3/12/2023	\$127.75	\$1,020.00
4/1/2024	\$127.75			
5/1/2024	\$127.75			
6/1/2024	\$127.75			
7/1/2024	\$127.75			
8/1/2024	\$127.75			
9/1/2024	\$127.75			
10/1/2024	\$127.75			
11/1/2024	\$127.75			

Item/Part	Qty	Amount	Total
Lumber	50	\$60.00	\$ 600.00
Roofing shingles	10	\$80.00	\$ 800.00

Subtotal	Rate(7.50%)	ESTR(50%)	Total
\$ 1,400.00	\$ 333.00	\$ 3,269.90	\$ 5,002.90
\$ 62.99		\$ 511.00	\$ 573.99
		\$ 1,084.99	\$ 1,658.98

FEATURE, FIX OR FOCUS

 *Excel For Freelancers*

Learn How To Create An Invoice With Installment Payments In Excel



DOWNLOAD APPLICATION



VIEW TRAINING

by: Randy Austin

ABOUT THE AUTHOR

A 4-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 471,000 YouTube subscribers, 35,448,742 video views, 430+ comprehensive training videos, and a thriving 65,000 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](#)



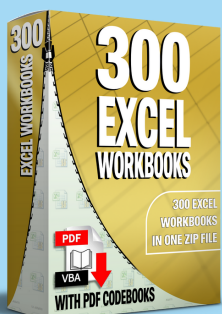
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



This incredible 13-hour freelancer masterclass will teach anyone how to be a successful freelancer with my proven 9-Phase 'Financial Freedom Roadmap' and includes 30+ downloads and exercises.

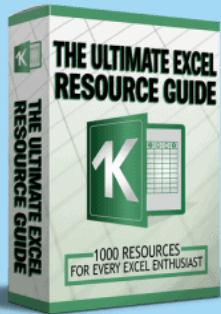
[Click here to learn more](#)



Incredible Package of 300 of my BEST Applications with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "300 Workbook Library".

[Click here to learn more](#)

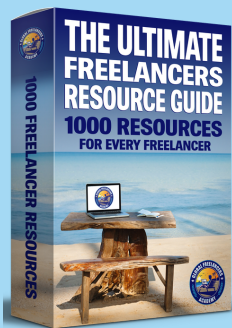
OUR COURSES & PRODUCTS



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](#)

-



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

[Click here to learn more](#)

-



Revolutionize the way you work with Excel and take productivity to the next level with the Excel AI Toolpack - the FIRST AI tool designed for ANY Windows Desktop version of Excel. This incredible add-in combines FIVE powerful AI tools, transforming your Excel into an intelligent powerhouse!

[Click here to learn more](#)

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Customers.....	3
(Declarations).....	3
InvItems.....	4
(Declarations).....	4
InvList.....	5
(Declarations).....	5
Invoice.....	6
(Declarations).....	6
Worksheet_Change [Sub].....	6
Worksheet_SelectionChange [Sub].....	7
InvPmnts.....	8
(Declarations).....	8
Items.....	9
(Declarations).....	9
Sheet1.....	10
(Declarations).....	10
ThisWorkbook.....	11
(Declarations).....	11
Modules.....	12
Invoice_Macros.....	12
(Declarations).....	12
CopyDownAmount [Sub].....	12
Invoice_Delete [Sub].....	12
Invoice_Email [Sub].....	12
Invoice_Load [Sub].....	13
Invoice_New [Sub].....	14
Invoice_Print [Sub].....	14
Invoice_SaveUpdate [Sub].....	14
Next_Invoice [Sub].....	15
Prev_Invoice [Sub].....	15
Pmnt_Macros.....	17
(Declarations).....	17
CreatePaymentSchedule [Sub].....	17
PmntItem_Delete [Sub].....	18
PmntItem_Load [Sub].....	18
PmntItem_LoadList [Sub].....	18
PmntItem_New [Sub].....	18
PmntItem_SaveUpdate [Sub].....	18
PopUp_Calendar.....	20
(Declarations).....	20
CalCol [Sub].....	20
CalendarHide [Sub].....	20
CalendarShow [Sub].....	20
CalFormulaReplacement [Sub].....	21
CheckForSheet [Sub].....	21
CreateCalSht [Sub].....	22
GroupCal [Sub].....	24
MacroLinkRemover [Sub].....	24
NextMonth [Sub].....	25
NextYear [Sub].....	25
PrevMonth [Sub].....	25
PrevYear [Sub].....	26
ReplaceCalendar [Sub].....	26
SelectDay [Sub].....	26
ShowSettings [Sub].....	26
UnGroupCal [Sub].....	27

1	Option Explicit
2	

1	Option Explicit
2	

1	Option Explicit
2	

1	Option Explicit
2	

```
1 Option Explicit
```

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

```
4 Dim CustRow As Long, ItemDBRow As Long
```

```
5 'On Invoice Customer Change, Load Customer Address & City, State & Zip
```

```
6 If Not Intersect(Target, Range("E3" )) Is Nothing And Range("E3" ).Value <> Empty
```

```
7 Then
```

```
8 If Range("B6" ).Value <> "" Then
```

```
9 CustRow = Range("B6" ).Value 'Customer Row
```

```
10 Range("K4" ).Value = Customers.Range("C" & CustRow).Value 'Address 1
```

```
11 Range("K5" ).Value = Customers.Range("D" & CustRow).Value & ", " & Customers
```

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

```
13 Else 'Clear Address Fields
```

```
14 Range("K4:L5" ).ClearContents
```

```
15 End If
```

```
16 End If
```

```
17 'On Invoice Item Change, but not on Invoice Load
```

```
18 If Not Intersect(Target, Range("K8:K34" )) Is Nothing And Range("B1" ).Value = False
```

```
19 Then
```

```
20 If Range("K" & Target.Row).Value <> Empty Then
```

```
21 On Error Resume Next
```

```
22 ItemDBRow = Items.Range("Item_Name" ).Find(Target.Value, , xlValues, xlWhole).
```

```
23 Row
```

```
24 On Error GoTo 0
```

```
25 If ItemDBRow = 0 Then
```

```
26 MsgBox "Item Not Found"
```

```
27 Exit Sub
```

```
28 End If
```

```
29 Range("L" & Target.Row).Value = Items.Range("C" & ItemDBRow).Value
```

```
30 'Description
```

```
31 Range("M" & Target.Row).Value = 1 'Set Default Qty to 1
```

```
32 Range("N" & Target.Row).Value = Items.Range("D" & ItemDBRow).Value 'Item
```

```
33 Amount
```

```
34 Else 'Clear Line Out
```

```
35 If Target.Row > 8 Then Range("L" & Target.Row & ":N" & Target.Row).
```

```
36 ClearContents 'Clear Item On Delete
```

```
37 End If
```

```
38 End If
```

```
39 'On Invoice Search Change
```

```
40 If Not Intersect(Target, Range("R2" )) Is Nothing And Range("R2" ).Value <> Empty
```

```
41 Then
```

```
42 If Range("B8" ).Value = Empty Then
```

```
43 MsgBox "Please enter a correct invoice #"
```

```
44 Exit Sub
```

```
45 End If
```

```
46 Range("B2" ).Value = Range("R2" ).Value 'Set Invoice #
```

```
47 Range("R2" ).ClearContents
```

```
48 Invoice_Load 'Load Invoice
```

```
49 End If
```

```
50 'On Interest Type Change
```

```
51 If Not Intersect(Target, Range("E9" )) Is Nothing Then
```

```
52 If Target.Value = "Fixed Amount Per Payment" Then
```

```
53 Range("H9" ).NumberFormat = "$0.00"
```

```
54 Else
```

```
55 Range("H9" ).NumberFormat = "0.00%"
```

```
56 End If
```

```
57 End If
```

```
1
```

```
1
53 End Sub
54
55 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
56     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
    it in
57     Dim Cal As Shape
58     'On Error Resume Next
59     Set Cal = Shapes("Calendar" )
60     On Error GoTo 0
61     If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
62     If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
63
64     If Target.Count > 1 Then Exit Sub
65     If Not Intersect(Target, Range("H3,H7,E13,E14,J8:J34,R6,T6" )) Is Nothing Then
        'Change this to any cell (or cells) you would like to have the Pop-Up Calendar
        Appear
66         CheckForSheet
67         CalendarShow
68     Else:
69         CheckForSheet
70         CalendarHide
71     End If
72     'On Selection Of Payment Item
73     If Not Intersect(Target, Range("D18:H9999" )) Is Nothing And Range("D" & Target.Row
        ).Value <> Empty Then
74         Range("B11" ).Value = Range("C" & Target.Row).Value 'Set Payment ID
75         PmntItem_Load 'Run Macro To load Payment Item
76     End If
77 End Sub
```

1 Option Explicit

2

1	Option Explicit
2	

1 Option Explicit

2

1 Option Explicit

2

```

1 Option Explicit
2 Dim InvRow As Long, InvCol As Long, ItemDBRow As Long, InvItemRow As Long, InvNumb As
Long
3 Dim LastRow As Long, LastItemRow As Long, LastResultRow As Long, ResultRow As Long
4
5 Sub CopyDownAmount()
6 Invoice.Range("H14").Value = Invoice.Range("H13").Value 'Copy Down To Payment
Amount
7 End Sub
8
9 Sub Invoice_Delete()
10 With Invoice
11 If MsgBox("Are you sure you want to delete this Invoice?" , vbYesNo, "Delete
Invoice" ) = vbNo Then Exit Sub
12 If .Range("B3").Value = Empty Then GoTo NotSaved
13 InvRow = .Range("B3").Value 'Set Invoice Row
14 InvList.Range(InvRow & ":" & InvRow).EntireRow.Delete
15 With InvItems
16 LastRow = .Range("A99999").End(xlUp).Row
17 If LastRow < 4 Then Exit Sub
18 .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
" N2:N3" ), CopyToRange:=.Range("P2:V2" ), Unique:=True
19 LastResultRow = .Range("V99999").End(xlUp).Row
20 If LastResultRow < 3 Then GoTo NoItems
21
22 For ResultRow = 3 To LastResultRow
23 ItemDBRow = .Range("V" & ResultRow).Value 'Set Invoice Database Row
24 .Range("A" & ItemDBRow & ":I" & ItemDBRow).ClearContents 'Clear Fields
(deleting creates issues with results
25 Next ResultRow
26 'Resort DB to remove spaces
27 With .Sort
28 .SortFields.Clear
29 .SortFields.Add Key:=InvItems.Range("A4" ), SortOn:=xlSortOnValues, Order:=
xlAscending, DataOption:=xlSortNormal 'Sort
30 .SetRange InvItems.Range("A4:I" & LastResultRow) 'Set Range
31 .Apply 'Apply Sort
32 End With
33 End With
34 NoItems:
35 NotSaved:
36 Invoice_New 'Add New Invoice
37 End With
38 End Sub
39
40 Sub Invoice_Email()
41 Dim OutMail As Object, OutApp As Object
42 Dim EmailTo As String, Subj As String, Mesg As String, FilePath As String
43 Dim CustRow As Long, VarRow As Long
44 With Invoice 'Check For Saved Invoice
45 If .Range("B3").Value = Empty Then
46 MsgBox "Please make sure to save the Invoice before emailing"
47 Exit Sub
48 End If
49 If .Range("B6").Value = Empty Then
50 MsgBox "Please make sure to select a customer before emailing"
51 Exit Sub
52 End If
53
54 'Set Variables

```



```

1 2
55 CustRow = .Range("B6" ).Value 'Customer Row
56 EmailTo = Customers.Range("H" & CustRow).Value 'Email Address
57 ' Subj = .Range("O6").Value 'Subject
58 ' Mesg = .Range("O7").Value 'Email Message
59 ' For VarRow = 6 To 13
60 '     Subj = Replace(Subj, .Range("Q" & VarRow).Value, Format(.Range("R" &
VarRow).Value, .Range("R" & VarRow).NumberFormat))
61 '     Mesg = Replace(Mesg, .Range("Q" & VarRow).Value, Format(.Range("R" &
VarRow).Value, .Range("R" & VarRow).NumberFormat))
62 '     Next VarRow
63 FilePath = ThisWorkbook.Path & "\" & Invoice.Range("E3" ).Value & "_" & Invoice
.Range("B2" ).Value & ".pdf" 'File Path (Customer Name & Inv. #)
64 If Dir(FilePath, vbDirectory) <> "" Then Kill (FilePath) 'Delete file if exists

65 .ExportAsFixedFormat xlTypePDF, FilePath, , , False, , , False
66 Set OutApp = CreateObject("Outlook.Application" ) 'Set Outlook Application
67 Set OutMail = OutApp.CreateItem(0) 'Create Email
68 With OutMail
69     .To = EmailTo 'Email address
70     .Subject = "" 'Subj
71     .body = "" 'Mesg
72     .attachments.Add FilePath
73     .display 'use .Send to send email directly
74 End With
75 End With
76 End Sub

77
78 Sub Invoice_Load()
79 With Invoice
80     If .Range("B3" ).Value = Empty Then
81         MsgBox "Please enter a correct Invoice #"
82         Exit Sub
83     End If
84     Application.ScreenUpdating = False
85     .Range("B1" ).Value = True 'Set Invoice Load to true
86     .Range("E3:F6,F7,E8:F8,H3,H7,H8,J8:N34,P8:P34,C18:G999" ).ClearContents
87     InvRow = .Range("B3" ).Value
88     For InvCol = 2 To 10
89         .Range(InvList.Cells(1, InvCol).Value).Value = InvList.Cells(InvRow, InvCol).
Value 'Load Invoice List Data
90     Next InvCol
91
92     'Load Invoice Items
93     With InvItems
94         LastRow = .Range("A99999" ).End(xlUp).Row
95         If LastRow < 4 Then Exit Sub
96         .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"N2:N3" ), CopyToRange:=.Range("P2:V2" ), Unique:=True
97         LastResultRow = .Range("V99999" ).End(xlUp).Row
98         If LastResultRow < 3 Then GoTo NoItems
99         For ResultRow = 3 To LastResultRow
100             InvItemRow = .Range("U" & ResultRow).Value 'Set Invoice Row
101             Invoice.Range("J" & InvItemRow & ":N" & InvItemRow).Value = .Range("P" &
ResultRow & ":T" & ResultRow).Value 'Item details
102             Invoice.Range("P" & InvItemRow).Value = .Range("V" & ResultRow).Value
'Set Item DB Row
103         Next ResultRow
104         NoItems:
105     End With
106 PmntItem_LoadList 'Run Macro To Load Payments

```

1 2

```

1 2
107 .Range("B1" ).Value = False 'Set Invoice Load To false
108 Application.ScreenUpdating = True
109 End With
110 End Sub
111 Sub Invoice_New()
112 With Invoice
113 .Range("E3:F6,F7,K4:L5,E8:F8,H3,H7,H8,J8:N34,P8:P34,C18:G999" ).ClearContents
114 .Range("B2" ).Value = .Range("B4" ).Value 'Set Next Invoice ID
115 .Range("H3" ).Value = Date 'Set Current Date Default
116 .Range("E4" ).Value = Admin.Range("J5" ).Value 'Invoice Type
117 .Range("F7" ).Value = Admin.Range("J6" ).Value 'Progress Payments
118 .Range("E8" ).Value = Admin.Range("J7" ).Value 'Frequency
119 .Range("H8" ).Value = Admin.Range("J8" ).Value '# Of Payments
120 End With
121 End Sub
122
123
124 Sub Invoice_Print()
125 Invoice.PrintOut , , , , True, , , , False
126 End Sub
127
128 Sub Invoice_SaveUpdate()
129 With Invoice
130 'Check For Required Fields
131 If .Range("E3" ).Value = Empty Then
132 MsgBox "Please make sure to add a customer before saving this invoice"
133 Exit Sub
134 End If
135 Application.ScreenUpdating = False
136 If .Range("B3" ).Value = Empty Then 'New Invoice
137 InvRow = InvList.Range("A99999" ).End(xlUp).Row + 1
138 InvList.Range("A" & InvRow).Value = .Range("B2" ).Value ' Next Invoice #
139 InvList.Range("M1:N1" ).Copy
140 InvList.Range("M" & InvRow & ":" & InvRow).PasteSpecial xlPasteFormulas
141 Application.CutCopyMode = False 'Turn Off
142 Else 'Existing Invoice
143 InvRow = .Range("B3" ).Value 'Set Existing Invoice Row
144 End If
145 For InvCol = 2 To 12
146 InvList.Cells(InvRow, InvCol).Value = .Range(InvList.Cells(1, InvCol).Value).
Value 'Save Invoice List Data
147 Next InvCol
148
149 'Save/Update Invoice Items
150 LastItemRow = .Range("K34" ).End(xlUp).Row
151 If LastItemRow < 8 Then GoTo NoItems
152 For InvItemRow = 8 To LastItemRow
153 If .Range("P" & InvItemRow).Value = "" Then 'New Inv. DB Item
154 ItemDBRow = InvItems.Range("A99999" ).End(xlUp).Row + 1
155 .Range("P" & InvItemRow).Value = ItemDBRow 'Set Item DB Row
156 InvItems.Range("A" & ItemDBRow).Value = .Range("B2" ).Value 'Invoice #
157 InvItems.Range("H" & ItemDBRow).Value = InvItemRow 'Set Invoice Row
158 InvItems.Range("I" & ItemDBRow).Value = "=Row()"
159 Else 'Existing Item
160 ItemDBRow = .Range("P" & InvItemRow).Value 'Invoice Item Row
161 End If
162 InvItems.Range("B" & ItemDBRow & ":" & ItemDBRow).Value = .Range("J" &
InvItemRow & ":" & InvItemRow).Value 'Save Invoice Item Details
163 Next InvItemRow
164 NoItems:

```

1 2

```

1 2
165 Application.ScreenUpdating = True
166 MsgBox "Invoice Saved"
167 End With
168 End Sub
169
170
171 Sub Next_Invoice()
172 With Invoice
173 Dim MaxInvNum As Long
174 On Error Resume Next
175 MaxInvNum = Application.WorksheetFunction.Max(InvList.Range("Inv_ID" ))
176 On Error GoTo 0
177 If MaxInvNum = 0 Then
178 MsgBox "Please create and save an Invoice first"
179 Exit Sub
180 End If
181 InvNum = .Range("N3" ).Value
182 If InvNum = 0 Or .Range("B3" ).Value = Empty Then 'On New Invoice
183 InvRow = InvList.Range("A4" ).Value 'On Empty Invoice Go to First one
created
184 Else 'On Existing Inv. find Previous one
185 InvRow = InvList.Range("Inv_ID" ).Find(InvNum, , xlValues, xlWhole).Row + 1
186 End If
187 If .Range("N3" ).Value >= MaxInvNum Then
188 MsgBox "You are at the last invoice"
189 Exit Sub
190 End If
191 .Range("B2" ).Value = InvList.Range("A" & InvRow).Value 'Place Inv. ID inside
cell
192 Invoice_Load
193 End With
194 End Sub
195
196
197
198 Sub Prev_Invoice()
199 With Invoice
200 Dim MinInvNum As Long
201 On Error Resume Next
202 MinInvNum = Application.WorksheetFunction.Min(InvList.Range("Inv_ID" ))
203 On Error GoTo 0
204 If MinInvNum = 0 Then
205 MsgBox "Please create and save an Invoice first"
206 Exit Sub
207 End If
208 InvNum = .Range("N3" ).Value
209 If InvNum = 0 Or .Range("B3" ).Value = Empty Then 'On New Invoice
210 InvRow = InvList.Range("A99999" ).End(xlUp).Row 'On Empty Invoice Go to last
one created
211 Else 'On Existing Inv. find Previous one
212 InvRow = InvList.Range("Inv_ID" ).Find(InvNum, , xlValues, xlWhole).Row - 1
213 End If
214 If .Range("N3" ).Value = 1 Or MinInvNum = 0 Or MinInvNum = .Range("N3" ).Value
Then
215 MsgBox "You are at the first invoice"
216 Exit Sub
217 End If
218 .Range("B2" ).Value = InvList.Range("A" & InvRow).Value 'Place Inv. ID inside
cell
219 Invoice_Load

```

```
220 | 1 2  
221 | End With  
    | End Sub
```

```

1 Option Explicit
2 Dim PmntRow As Long, PmntCol As Long, LastRow As Long, LastResultRow As Long, SelRow As
3 Long, InvRow As Long
4 Dim ServItem As String
5
6 Sub CreatePaymentSchedule()
7     Dim Freq As String
8     Dim InvTotal As Double, PmntAmt As Double
9     Dim PmntNum As Long, NumOfPmnts As Long
10    Dim StartDt As Date, DueDt As Date
11    With Invoice
12        If .Range("B3").Value = Empty Then
13            MsgBox "Please make sure to save the Invoice before creating a payment
14            schedule"
15            Exit Sub
16        End If
17        If .Range("H4").Value = 0 Or .Range("H6").Value = 0 Then
18            MsgBox "Please make sure the Invoice Total and Balance are above 0"
19            Exit Sub
20        End If
21        If .Range("B9").Value < 3 Or .Range("F7").Value <> "Yes" Then
22            MsgBox "Please make sure to add in a Start On Date, Frequency and # of
23            Payments and set 'Use Progress Payments' to 'Yes' before creating a payment
24            schedule"
25            Exit Sub
26        End If
27        Application.ScreenUpdating = False
28        Freq = .Range("E8").Value 'Set Frequency
29        StartDt = .Range("H7").Value 'Start on Date
30        DueDt = StartDt 'Set First Due Date
31        InvTotal = .Range("H4").Value 'Balance Due
32        PmntItem_New 'Clear Payment Fields
33        NumOfPmnts = .Range("H8").Value 'Number Of Payments
34        PmntAmt = InvTotal / NumOfPmnts 'Set Payment Amount
35        If .Range("E9").Value <> "Do Not Charge Interest" Then 'Update Payment Amount
36            to include interest
37            PmntAmt = PmntAmt + (.Range("B10").Value / NumOfPmnts) 'Payment Amount +
38            Total Interest Divided By # of Payments
39        End If
40
41        .Range("B16").Value = True 'Set Create Pay Schedule To True
42        For PmntNum = 1 To NumOfPmnts
43            PmntItem_New 'Clear Form to Add New Payments
44            .Range("E13").Value = DueDt 'Set Due Date
45            .Range("H13").Value = PmntAmt 'Payment Amount
46            PmntItem_SaveUpdate 'Run Macro to Save Payment
47            Select Case Freq 'Update Due Date
48                Case Is = "Daily"
49                    DueDt = DateAdd("d", 1, DueDt)
50                Case Is = "Weekly"
51                    DueDt = DateAdd("ww", 1, DueDt)
52                Case Is = "Bi-Weekly"
53                    DueDt = DateAdd("ww", 2, DueDt)
54                Case Is = "Monthly"
55                    DueDt = DateAdd("m", 1, DueDt)
56                Case Is = "Semi-Monthly"
57                    DueDt = DateAdd("m", 0.5, DueDt)
58                Case Is = "Bi-Monthly"
59                    DueDt = DateAdd("m", 2, DueDt)

```

1 2 3 4

```

1 2 3 4
55     Case Is = "Annually"
56         DueDt = DateAdd("yyyy" , 1, DueDt)
57     End Select
58     Next PmntNumb
59     .Range("B16" ).Value = False 'Set Create Pay Schedule To False
60     PmntItem_New 'Clear Payment Fields
61     PmntItem_LoadList 'Load List
62     Application.ScreenUpdating = True
63 End With
64 End Sub

65
66 Sub PmntItem_Delete()
67     If MsgBox("Are you sure you want to delete this Payment Items?" , vbYesNo, "Delete
        Payment Items" ) = vbNo Then Exit Sub
68     With Invoice
69         If .Range("B13" ).Value = "" Then GoTo NotSaved
70         PmntRow = .Range("B13" ).Value 'Entry Row
71         InvPmnts.Range(PmntRow & ":" & PmntRow).EntireRow.Delete
72         NotSaved:
73         PmntItem_New
74         PmntItem_LoadList
75     End With
76 End Sub

77
78 Sub PmntItem_Load()
79     With Invoice
80         If .Range("B13" ).Value = "" Then
81             MsgBox "Please select a correct Payment to load"
82             Exit Sub
83         End If
84         PmntRow = .Range("B13" ).Value 'Set Entry Row
85         For PmntCol = 5 To 8 'Not On Project, Customer Or Service ID
86             .Range(InvPmnts.Cells(1, PmntCol).Value).Value = InvPmnts.Cells(PmntRow,
                PmntCol).Value 'Load Data
87         NextCol:
88     Next PmntCol
89 End With
90 End Sub

91 Sub PmntItem_LoadList()
92     Invoice.Range("C18:G999" ).ClearContents
93     With InvPmnts
94         LastRow = .Range("A99999" ).End(xlUp).Row
95         If LastRow < 4 Then Exit Sub
96         .Range("A3:M" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
            "M2:M3" ), CopyToRange:=.Range("Q2:V2" ), Unique:=True
97         LastResultRow = .Range("Q99999" ).End(xlUp).Row
98         If LastResultRow < 3 Then Exit Sub
99         Invoice.Range("C18:G" & LastResultRow + 15).Value = .Range("Q3:V" &
            LastResultRow).Value 'Bring Over Payment Items Results
100    End With
101 End Sub

102 Sub PmntItem_New()
103     Invoice.Range("E13,E14,H13,H14" ).ClearContents 'Clear Payment Fields
104     Invoice.Range("B11" ).Value = Invoice.Range("B14" ).Value 'Set Next Payment ID
105 End Sub

106
107 Sub PmntItem_SaveUpdate()
108     With Invoice
109         'Check For Required Fields
110         If .Range("E13" ).Value = "" Or .Range("H13" ).Value = Empty Then

```

```
111     1 2 3     MsgBox "Please make sure to enter a Due Date and Scheduled Payment Amount  
112         before saving"  
113     Exit Sub  
114     End If  
115     If .Range("B13" ).Value = Empty Then 'New Payment  
116         PmntRow = InvPmnts.Range("A99999" ).End(xlUp).Row + 1 'First avail row  
117         InvPmnts.Range("A" & PmntRow).Value = .Range("B11" ).Value 'Set Payment ID  
118         InvPmnts.Range("I" & PmntRow).Value = "=Row()"  
119     Else 'Existing Payment  
120         PmntRow = .Range("B13" ).Value  
121     End If  
122     For PmntCol = 2 To 8  
123         InvPmnts.Cells(PmntRow, PmntCol).Value = .Range(InvPmnts.Cells(1, PmntCol).  
124             Value).Value 'Save Data  
125     Next PmntCol  
126     If .Range("B16" ).Value = False Then 'Only Reload List & Set Message box when  
127         Creating Payment Schedule is False  
128         PmntItem_LoadList  
129         MsgBox "Payment Saved"  
130     End If  
131 End With  
132 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        On Error Resume Next
13        ActiveSheet.Range("A7" ).Select
14        On Error GoTo 0
15    End With
16 End Sub
17 Sub CalendarHide()
18     Dim DayNum As Long
19     'Hide Calendar, Reset Day Colors
20     On Error GoTo NoCal
21     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
22     Sheets("CalPopUp" ).Range("A7" ).Value = ""
23     If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
24         For DayNum = 1 To 42
25             DayName = DayNum & "Day"
26             With ActiveSheet.Shapes(DayName)
27                 .Fill.ForeColor.RGB = RGB(255, 255, 255)
28                 .TextFrame2.TextRange.Font.Bold = msoFalse
29             End With
30         Next DayNum
31     End If
32     Exit Sub
33     NoCal: 'If calendar has been removed by accident, paste in backup calendar from
34         CalPopUp Sheet
35 End Sub
36 Sub CalendarShow()
37     With ActiveSheet
38         Set SelCell = ActiveCell
39         'Check if active cell is a valid date
40         If IsDate(SelCell.Value) = True Then
41             Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
42         Else: 'If No Date or incorrect Date user current date
43             Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
44         End If
45         'Clear all shapes to white (if calendar is visible)
46         If ActiveSheet.Shapes("Calendar" ).Visible = True Then
47             For DayNum = 1 To 42
48                 DayName = DayNum & "Day"
49                 With ActiveSheet.Shapes(DayName)
50                     .Fill.ForeColor.RGB = RGB(255, 255, 255)
51                     .TextFrame2.TextRange.Font.Bold = msoFalse
52                 End With
53             Next DayNum
54         End If
55
56         Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).
57         Value) 'Set Month
58         Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).
59         Value) 'Set Year

```

1 2


```

1 2
58 DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
59 ' UnGroupCal
60 If InStr(.Shapes("1Day").DrawingObject.Formula, "]" ) <> 0 Then 'Run Workbook
    Link Remover and Cell Link Replacement
61     MacroLinkRemover
62     CalFormulaReplacement
63 End If
64 'GroupCal
65 On Error GoTo NoCal
66 If DayName = "Day" Then DayName = Day(Date) & "Day" 'Set Default Day
67 With ActiveSheet.Shapes(DayName)
68     .Fill.ForeColor.RGB = RGB(252, 213, 180)
69     .TextFrame2.TextRange.Font.Bold = msoTrue
70 End With
71 On Error GoTo NoCal
72 .Shapes("Calendar").Visible = msoCTrue
73 .Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3", "CalCol4",
    "CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible = False
74 .Shapes("Calendar").Left = SelCell.Left
75 .Shapes("Calendar").Placement = xlMove
76 .Shapes("Calendar").ZOrder msoBringToFront
77 .Shapes("Calendar").Top = SelCell.Offset(1, 0).Top
78 If Sheets("CalPopUp").Range("A6").Value > 0 Then
79     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
    "42Day")).Visible = True
80 Else:
81     .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
    "42Day")).Visible = False
82 End If
83 Sheets("CalPopUp").Range("A7").Value = SelCell.Address
84 ActiveCell.Select
85 End With
86 Exit Sub
87 NoCal:
88 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the
    calendar over from another sheet and paste into this sheet"
89 End Sub
90 Sub CalFormulaReplacement()
91     With ActiveSheet
92         Dim DayNum, ColNum, RowNum As Long
93         Dim Shp As Shape
94         ColNum = 2
95         RowNum = 1
96         For DayNum = 1 To 42
97             .Shapes(DayNum & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(RowNum
    , ColNum).Address
98             ColNum = ColNum + 1
99             If ColNum = 9 Then
100                 ColNum = 2
101                 RowNum = RowNum + 1
102             End If
103         Next DayNum
104         .Shapes("Month").DrawingObject.Formula = "=CalPopUp!$A$4"
105         .Shapes("Year").DrawingObject.Formula = "=CalPopUp!$A$2"
106     End With
107 End Sub
108
109 Sub CheckForSheet()
110     'Checks for existence of Calendar Pop-up Worksheet
111     Dim ws As Worksheet

```

```

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

```

1 2

1 2

```

168 .Range("A5" ).Value = "=A4&" & Chr(34) & " " & Chr(34) & "&CalYear"
169 .Range("A6" ).Value = "=SUM(B6:H6)"
170 .Range("A8" ).Value = "January"
171 .Range("A8" ).AutoFill Destination:=.Range("A8:A19" ), Type:=xlFillDefault
172 .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) & ")"
173 'Set Defined Names
174 ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
175 ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
176
177 'Add in Calendar Formulas
178
179 .Range("B1" ).Value =
"=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,Ca
lMonths,0),1)," & Chr(34) & Chr(34) & ")"
180 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
181 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
182 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
183 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
184 .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
185 .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($
$4,CalMonths,0),1)," & Chr(34) & Chr(34) & ")"
186 .Range("B2" ).Value = "=H1+1"
187 .Range("C2" ).Value = "=B2+1"
188 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
189 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
190 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
& Chr(34) & Chr(34) & ",H4+1)"
191 .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) &
",B5+1)," & Chr(34) & Chr(34) & ")"
192 .Range("B6" ).Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) &
",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) &
")"
193 .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) &
",I5+1)," & Chr(34) & Chr(34) & ")"
194 .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
195
196 'Set format to Single Day
197 .Range("B1:H6" ).NumberFormat = "d"
198
199 'Add in relative Day #'s
200 .Range("B7" ).Value = "1"
201 .Range("C7" ).Value = "2"
202 .Range("B8" ).Value = "8"
203 .Range("C8" ).Value = "9"
204 .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
205 .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
206 GroupCal

```

1 2

```

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

```

```

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

```

```

1
308 'Previous Month Button
309 If Sheets("CalPopUp").Range("A20").Value <> Empty Then
310     DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
311     With ActiveSheet.Shapes(DayName)
312         .Fill.ForeColor.RGB = RGB(255, 255, 255)
313         .TextFrame2.TextRange.Font.Bold = msoFalse
314     End With
315 End If
316 With Sheets("CalPopUp")
317     If .Range("A3").Value = 1 Then
318         .Range("A3").Value = 12
319         .Range("A2").Value = .Range("A2").Value - 1
320     Else:
321         .Range("A3").Value = .Range("A3").Value - 1
322     End If
323     If .Range("A6").Value > 0 Then
324         ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day",
325             "40Day", "41Day", "42Day")).Visible = True
326     Else:
327         ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day",
328             "40Day", "41Day", "42Day")).Visible = False
329     End If
330 End With
331 End Sub
332
333 Sub PrevYear()
334     ThisWorkbook.Sheets("CalPopUp").Range("A2").Value = ThisWorkbook.Sheets(
335         "CalPopUp").Range("A2").Value - 1
336 End Sub
337
338 Sub ReplaceCalendar() 'Shape Deleted
339     ThisWorkbook.Sheets("CalPopUp").Shapes("Calendar").Copy 'Copy From Developers
340     sheet
341     ActiveCell.Select 'Select the active Cell
342     ActiveSheet.Paste 'Paste in Calendar
343 End Sub
344
345 .....
346 ''''Select Day Of The Month
347 .....
348 Sub SelectDay()
349     Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
350     DayNumb = Replace(Application.Caller, "Day", "")
351     RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
352     ColNumb = DayNumb Mod 7 + 1
353     If ColNumb = 1 Then ColNumb = 8
354     'On Error Resume Next
355     If ThisWorkbook.Sheets("CalPopUp").Range("A7").Value = Empty Then Exit Sub
356     ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp").Range("A7").Value).Value =
357         ThisWorkbook.Sheets("CalPopUp").Cells(RowNumb, ColNumb).Value
358     ActiveSheet.Shapes("Calendar").Visible = msoFalse
359     ActiveCell.Offset(0, 1).Select
360 End Sub
361
362 Sub ShowSettings()
363     'Show or Hide Calendar Settings Panel
364     If ActiveSheet.Shapes.Range(Array("Settings")).Visible = True Then
365         ActiveSheet.Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3",
366             "CalCol4", "CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible
367         = False
368     Else:
369         ActiveSheet.Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3",

```

```
1 2  
    "CalCol14" , "CalCol15" , "CalCol16" , "CalCol17" , "CalCol18" , "CalCol19" ))).Visible  
    = True  
361 End If  
362 End Sub  
363  
364 Sub UnGroupCal()  
365     On Error Resume Next  
366     ActiveSheet.Shapes("Calendar" ).Ungroup  
367     ActiveSheet.Shapes("NextMonth" ).Ungroup  
368     ActiveSheet.Shapes("PrevMonth" ).Ungroup  
369     On Error GoTo 0  
370 End Sub
```

—, 24

A

Activate, 22
 ActiveCell, 20, 21, 26
 ActiveSheet, 20-22, 24-27
 ActiveWorkbook, 22, 23
 ActSht, 22
 Add, 12, 13, 22, 23
 Address, 21, 22
 Admin, 14
 AdvancedFilter, 12, 13, 18
 After, 22
 Application, 13-15, 17, 18, 20, 26
 Apply, 12
 Array, 20, 21, 24-26
 attachments, 13
 AutoFill, 23

B

body, 13
 Bold, 20, 21, 25, 26

C

Cal, 7
 CalCol, 20, 22
 CalendarHide, 7, 20
 CalendarShow, 7, 20
 CalFormulaReplacement, 21
 Caller, 20, 26
 Cells, 13, 14, 18, 19, 21, 22, 26
 CheckForSheet, 7, 21
 Chr, 23
 Clear, 12
 ClearContents, 6, 12-14, 18
 ColCnt, 22
 ColNum, 21
 ColNumb, 26
 Copy, 14, 22, 26
 CopyDownAmount, 12
 CopyToRange, 12, 13, 18
 Count, 7, 22
 CreateCalSht, 22
 CreateItem, 13
 CreateObject, 13
 CreatePaymentSchedule, 17
 CreateWS, 22
 CriteriaRange, 12, 13, 18
 Customers, 6, 13
 CustRow, 6, 12, 13
 CutCopyMode, 14

D

DataOption, 12
 DateAdd, 17, 18
 Day, 21
 DayCnt, 22
 DayName, 20, 21, 25, 26
 DayNum, 20, 21
 DayNumb, 26
 Delete, 12, 18
 Destination, 23
 Dir, 13
 display, 13
 DrawingObject, 21, 22
 DueDt, 17, 18

E

EmailTo, 12, 13
 Empty, 6, 7, 12-15, 17-20, 25, 26
 EntireRow, 12, 18
 Explicit, 2-6, 8-12, 17
 ExportAsFixedFormat, 13

F

FilePath, 12, 13
 Fill, 20, 21, 25, 26
 Find, 6, 15
 Font, 20, 21, 25, 26
 ForeColor, 20, 21, 25, 26
 Formula, 21, 22
 Freq, 17

G

Group, 24
 GroupCal, 23, 24

I

InStr, 21, 25
 Intersect, 6, 7
 InvCol, 12-14
 InvItemRow, 12-14
 InvItems, 12-14
 InvList, 12-15
 InvNumb, 12, 15
 Invoice, 12-15, 17, 18
 Invoice_Delete, 12
 Invoice_Email, 12
 Invoice_Load, 6, 13, 15
 Invoice_New, 12, 14
 Invoice_Print, 14
 Invoice_SaveUpdate, 14
 InvPmnts, 18, 19
 InvRow, 12-15, 17
 InvTotal, 17
 IsDate, 20
 ItemDBRow, 6, 12, 14
 Items, 6

K

Key, 12
 Kill, 13

L

LastItemRow, 12, 14
 LastResultRow, 12, 13, 17, 18
 LastRow, 12, 13, 17, 18
 Left, 21, 25
 Len, 25

M

MacroLink, 25
 MacroLinkRemover, 21, 24
 Max, 15
 MaxInvNumb, 15
 Mesg, 12
 Min, 15
 MinInvNumb, 15
 Month, 20, 22
 MsgBox, 6, 12-15, 17-19, 21, 24
 msoBringToFront, 21
 msoCTrue, 21, 24
 msoFalse, 20, 25, 26
 msoTrue, 21

N

Name, 22-24
 Names, 23
 NewLink, 25
 Next_Invoice, 15
 NextCol, 18
 NextMonth, 25
 NextShp, 25
 NextYear, 25
 NoCal, 20, 21, 24
 Noltems, 12-14
 NotSaved, 12, 18
 NumberFormat, 6, 23
 NumbOfPmnts, 17

O

Offset, 21, 26
 OnAction, 22, 25
 Order, 12
 OutApp, 12, 13
 OutMail, 12, 13

P

Paste, 22, 26
 PasteSpecial, 14
 Path, 13
 Placement, 21, 24
 PmntAmnt, 17
 PmntCol, 17-19
 PmntItem_Delete, 18
 PmntItem_Load, 7, 18
 PmntItem_LoadList, 13, 18, 19
 PmntItem_New, 17, 18
 PmntItem_SaveUpdate, 17, 18
 PmntNumb, 17, 18
 PmntRow, 17-19
 Prev_Invoice, 15
 PrevMonth, 25
 PrevYear, 26
 PrintOut, 14

R

Range, 6, 7, 12-15, 17-26
 RefersTo, 23
 Replace, 26
 ReplaceCalendar, 7, 26
 ResultRow, 12, 13
 RGB, 20, 21, 25, 26
 Right, 25
 RoundUp, 26
 Row, 6, 7, 12-15, 18, 19
 RowCnt, 22
 RowNum, 21
 RowNumb, 26

S

ScreenUpdating, 13-15, 17, 18
 SelCell, 20, 21
 SelectDay, 26
 Selection, 20, 24
 SelRow, 17
 ServItem, 17
 SetRange, 12
 Shape, 7, 21, 25
 ShapeRange, 20, 24
 Shapes, 7, 20-22, 24-27
 Sheets, 20-22, 25, 26
 ShowSettings, 26
 Shp, 21, 25

Sort, [12](#)
SortFields, [12](#)
SortOn, [12](#)
Split, [25](#)
SplitLink, [25](#)
StartDt, [17](#)
Subj, [12](#)
Subject, [13](#)

T

Target, [6](#), [7](#)
TextFrame2, [20](#), [21](#), [25](#), [26](#)
TextRange, [20](#), [21](#), [25](#), [26](#)
ThisWorkbook, [13](#), [22](#), [25](#), [26](#)
Top, [21](#)

U

Ungroup, [27](#)
UnGroupCal, [22](#), [27](#)
Unique, [12](#), [13](#), [18](#)

V

Value, [6](#), [7](#), [12-15](#), [17-23](#), [25](#), [26](#)
VarRow, [12](#)
vbDirectory, [13](#)
vbNo, [12](#), [18](#)
vbYesNo, [12](#), [18](#)
Visible, [7](#), [20-22](#), [24-27](#)

W

Worksheet, [21](#), [22](#)
Worksheet_Change, [6](#)
Worksheet_SelectionChange, [7](#)
WorksheetFunction, [15](#), [26](#)
ws, [21](#), [22](#)

X

xlAscending, [12](#)
xlFillDefault, [23](#)
xlFilterCopy, [12](#), [13](#), [18](#)
xlMove, [21](#), [24](#)
xlPasteFormulas, [14](#)
xlSheetHidden, [22](#)
xlSortNormal, [12](#)
xlSortOnValues, [12](#)
xlTypePDF, [13](#)
xlUp, [12-15](#), [18](#), [19](#)
xlValues, [6](#), [15](#)
xlWhole, [6](#), [15](#)

Y

Year, [20](#), [22](#)

Z

ZOrder, [21](#)

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