

VBA SOURCE CODE BOOK

CUSTOMER STATEMENTS UPDATED

EXCLUSIVE FOR SILVER & GOLD MEMBERS

Order Date	Order #	Order Type	Due Date	Order Total	Payments	Balance
4/9/2022	4	Invoice				\$347.00
6/9/2022	8	Invoice				\$807.00
6/12/2022	9	Invoice				\$506.00
6/10/2022	18	Invoice				\$431.00
6/11/2022	19	Invoice	7/4/2022	42	Invoice	\$431.00
6/12/2022	20	Invoice	7/5/2022	43	Invoice	\$431.00
6/13/2022	21	Invoice	7/6/2022	44	Invoice	\$431.00
6/14/2022	22	Invoice	7/7/2022	45	Invoice	\$431.00
6/15/2022	23	Invoice	7/8/2022	46	Invoice	\$431.00
6/16/2022	24	Invoice	7/9/2022	47	Invoice	\$431.00
6/17/2022	25	Invoice	7/10/2022	48	Invoice	\$431.00
6/18/2022	26	Invoice	7/11/2022	49	Invoice	\$431.00
6/19/2022	27	Invoice	7/12/2022	50	Invoice	\$431.00
6/20/2022	28	Invoice	7/13/2022	51	Invoice	\$431.00
6/21/2022	29	Invoice	7/14/2022	52	Invoice	\$431.00
6/22/2022	30	Invoice	7/15/2022	53	Invoice	\$431.00
6/23/2022	31	Invoice	7/16/2022	54	Invoice	\$431.00
6/24/2022	32	Invoice	7/17/2022	55	Invoice	\$431.00
6/25/2022	33	Invoice	7/18/2022	56	Invoice	\$431.00
6/26/2022	34	Invoice	7/19/2022	57	Invoice	\$431.00
6/27/2022	35	Invoice	7/20/2022	58	Invoice	\$431.00
6/28/2022	36	Invoice	7/21/2022	59	Invoice	\$431.00
6/29/2022	37	Invoice	7/22/2022	60	Invoice	\$431.00
6/30/2022	38	Invoice	7/23/2022	61	Invoice	\$431.00
7/1/2022	39	Invoice	7/24/2022	62	Invoice	\$431.00
7/2/2022	40	Invoice	7/25/2022	63	Invoice	\$431.00
7/3/2022	41	Invoice	7/26/2022	64	Invoice	\$431.00
7/4/2022	42	Invoice	7/27/2022	65	Invoice	\$431.00
7/5/2022	43	Invoice	7/28/2022	66	Invoice	\$431.00
7/6/2022	44	Invoice	7/29/2022	67	Invoice	\$431.00
7/7/2022	45	Invoice	7/30/2022	68	Invoice	\$431.00
7/8/2022	46	Invoice	7/31/2022	69	Invoice	\$431.00
7/9/2022	47	Invoice	8/1/2022	70	Invoice	\$431.00
7/10/2022	48	Invoice	8/2/2022	71	Invoice	\$431.00
7/11/2022	49	Invoice	8/3/2022	72	Invoice	\$431.00
7/12/2022	50	Invoice	8/4/2022	73	Invoice	\$431.00
7/13/2022	51	Invoice	8/5/2022	74	Invoice	\$431.00
7/14/2022	52	Invoice	8/6/2022	75	Invoice	\$431.00
7/15/2022	53	Invoice	8/7/2022	76	Invoice	\$431.00
7/16/2022	54	Invoice	8/8/2022	77	Invoice	\$431.00
7/17/2022	55	Invoice	8/9/2022	78	Invoice	\$431.00
7/18/2022	56	Invoice	8/10/2022	79	Invoice	\$431.00
7/19/2022	57	Invoice	8/11/2022	80	Invoice	\$431.00
7/20/2022	58	Invoice	8/12/2022	81	Invoice	\$431.00
7/21/2022	59	Invoice	8/13/2022	82	Invoice	\$431.00
7/22/2022	60	Invoice	8/14/2022	83	Invoice	\$431.00
7/23/2022	61	Invoice	8/15/2022	84	Invoice	\$431.00
7/24/2022	62	Invoice	8/16/2022	85	Invoice	\$431.00
7/25/2022	63	Invoice	8/17/2022	86	Invoice	\$431.00
7/26/2022	64	Invoice	8/18/2022	87	Invoice	\$431.00
7/27/2022	65	Invoice	8/19/2022	88	Invoice	\$431.00
7/28/2022	66	Invoice	8/20/2022	89	Invoice	\$431.00

FEATURE, FIX, OR FOCUS



Excel For Freelancers

Automatically Create & Send Customers Statements 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



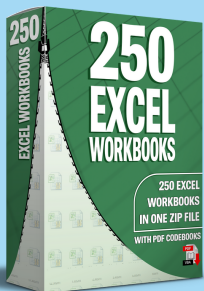
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



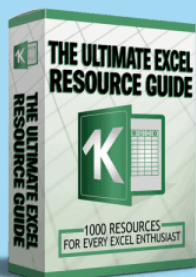
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



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

[Click here to learn more](#)



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

[Click here to learn more](#)

Projects.....	3
VBAProject.....	3
Documents.....	3
Admin.....	3
(Declarations).....	3
Worksheet_Change [Sub].....	3
Worksheet_SelectionChange [Sub].....	3
Dashboard.....	5
(Declarations).....	5
Worksheet_Change [Sub].....	5
Home.....	6
(Declarations).....	6
Items.....	7
(Declarations).....	7
Worksheet_Change [Sub].....	7
Worksheet_SelectionChange [Sub].....	7
ItemsDB.....	8
(Declarations).....	8
NamesDB.....	9
(Declarations).....	9
OrderItemsDB.....	10
(Declarations).....	10
Orders.....	11
(Declarations).....	11
Worksheet_Change [Sub].....	11
Worksheet_SelectionChange [Sub].....	12
OrdersDB.....	13
(Declarations).....	13
ThisWorkbook.....	14
(Declarations).....	14
Transactions.....	15
(Declarations).....	15
Worksheet_Change [Sub].....	15
Worksheet_SelectionChange [Sub].....	15
TransDB.....	16
(Declarations).....	16
Forms.....	17
AddEditName.....	17
(Declarations).....	17
CancelBtn_Click [Sub].....	17
SaveBtn_Click [Sub].....	17
Modules.....	18
App_Macros.....	18
(Declarations).....	18
BrowseForPicFolder [Sub].....	18
BuildListOfAccounts [Sub].....	18
Menu_GoToSheet [Sub].....	18
Dashboard_Macros.....	19
(Declarations).....	19
Dashboard_Refresh [Sub].....	19
Item_Macros.....	21
(Declarations).....	21
Item_BrowseItemPic [Sub].....	21
Item_ClearItemPic [Sub].....	21
Item_Delete [Sub].....	21
Item_Load [Sub].....	21
Item_New [Sub].....	22
Item_SavedMsg [Sub].....	22
Item_SaveUpdate [Sub].....	22
Item_ShowItemPic [Sub].....	23
Items_UpdateSortedNames [Sub].....	23
Names_Macros.....	24
(Declarations).....	24
Names_UpdateAndSort [Sub].....	24
Order_Macros.....	26
(Declarations).....	26

Order_CopyTo [Sub].....	26
Order_Delete [Sub].....	27
Order_Email [Sub].....	27
Order_Load [Sub].....	28
Order_New [Sub].....	29
Order_Print [Sub].....	30
Order_SavedMsg [Sub].....	30
Order_SaveUpdate [Sub].....	31
OrderSheet_Macros.....	33
(Declarations).....	33
Order_AddNewMenu [Sub].....	33
Order_CopyToMenu [Sub].....	33
Orders_AddEditName [Sub].....	33
StatementMacros.....	34
(Declarations).....	34
Statement_Create [Sub].....	34
Statement_CreateAll [Sub].....	34
Statement_Email [Sub].....	35
Statement_Preview [Sub].....	36
Statement_PrintOut [Sub].....	36
Trans_Macros.....	37
(Declarations).....	37
Trans_AddEditName [Sub].....	37
Trans_ClearFilter [Sub].....	37
Trans_Delete [Sub].....	37
Trans_Load [Sub].....	37
Trans_LoadList [Sub].....	38
Trans_New [Sub].....	38
Trans_SavedMsg [Sub].....	38
Trans_SaveUpdate [Sub].....	38

```
1 Option Explicit
```

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

```
4 'On any Income Or Expense change, rebuild all accounts (Sorted) List
```

```
5 If Not Intersect(Target, Range("B18:C39" )) Is Nothing Then BuildListOfAccounts
```

```
7 'Make Database updates on order type name change
```

```
8 If Not Intersect(Target, Range("C5:C8" )) Is Nothing And Range("AB" & Target.Row).  
Value <> Range("C" & Target.Row).Value Then
```

```
9 Dim OrderType As String, OldType As String
```

```
10 Dim OrderRow As Long, LastRow As Long
```

```
11 OldType = Range("AB" & Target.Row).Value 'Old Type
```

```
12 OrderType = Target.Value 'New Type
```

```
13 'Replay Order types in Order Database
```

```
14 With OrdersDB
```

```
15 LastRow = .Range("A99999" ).End(xlUp).Row
```

```
16 For OrderRow = 4 To LastRow
```

```
17 If .Range("B" & OrderRow).Value = OldType Then .Range("B" & OrderRow).  
Value = OrderType 'Set to new
```

```
18 Next OrderRow
```

```
19 End With
```

```
20 'Replace Types in Order Items DB
```

```
21 With OrderItemsDB
```

```
22 LastRow = .Range("A99999" ).End(xlUp).Row
```

```
23 For OrderRow = 4 To LastRow
```

```
24 If .Range("B" & OrderRow).Value = OldType Then .Range("B" & OrderRow).  
Value = OrderType 'Set to new
```

```
25 Next OrderRow
```

```
26 End With
```

```
27 End If
```

```
28 End Sub
```

```
30 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
31 Dim Cell As Range
```

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

```
34 'Default Estimate, Work Order, Invoice, Purchase Order Status
```

```
35 If Not Intersect(Target, Range("H5:H11,K5:K11,N5:N11,Q5:Q11" )) Is Nothing And  
Target.Offset(0, -1).Value <> Empty Then
```

```
36 Set Cell = Target
```

```
37 Range(Cells(5, Target.Column), Cells(11, Target.Column)).ClearContents
```

```
38 If Cell.Value = Chr(252) Then Cell.ClearContents Else Cell.Value = Chr(252)
```

```
39 Target.Offset(0, -1).Select
```

```
40 End If
```

```
43 'Set Default Payment Terms
```

```
44 If Not Intersect(Target, Range("N15:N21" )) Is Nothing And Range("L" & Target.Row).  
Value <> Empty Then
```

```
45 Set Cell = Target
```

```
46 Range("N15:N21" ).ClearContents
```

```
47 If Cell.Value = Chr(252) Then Cell.ClearContents Else Cell.Value = Chr(252)
```

```
48 Range("L15" ).Select
```

```
49 End If
```

```
50 If Not Intersect(Target, Range("C5:C8" )) Is Nothing Then
```

```
51 Range("AB" & Target.Row).Value = Target.Value 'Set Prev string
```

```
52 End If
```

```
55 'Set Default Footer messages
```

```
56 If Not Intersect(Target, Range("R28:V39" )) Is Nothing And Range("F" & Target.Row).  
Value <> Empty Then
```

```
1 2  
Value <> Empty Then  
57   Set Cell = Target  
58   Range(Cells(28, Target.Column), Cells(39, Target.Column)).ClearContents  
59   If Cell.Value = Chr(252) Then Cell.ClearContents Else Cell.Value = Chr(252)  
60   Range("F" & Target.Row).Select  
61   End If  
62 End Sub
```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("E3" )) Is Nothing Then
5         Dim DtRange As String
6         Dim RangeRow As Long
7         Range("AA1" ).Value = True
8         DtRange = Target.Value
9         On Error Resume Next
10        RangeRow = Admin.Range("DateRange" ).Find(DtRange, , xlValues, xlWhole).Row
11        On Error GoTo 0
12        If RangeRow <> 0 Then 'Set From and To Dates
13            Range("H3" ).Value = Admin.Range("Q" & RangeRow).Value 'Set From Date
14            Range("K3" ).Value = Admin.Range("S" & RangeRow).Value 'Set To Date
15            Calculate
16        End If
17        Range("AA1" ).Value = False
18        Dashboard_Refresh
19    End If
20
21    If Not Intersect(Target, Range("H3,K3" )) Is Nothing And Range("H3" ).Value <> Empty
22        And Range("K3" ).Value <> Empty And Range("AA1" ).Value = False Then
23        Dashboard_Refresh
24
25 End Sub
```


1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5     If Not Intersect(Target, Range("E3" )) Is Nothing And Range("E3" ).Value <> "" Then
6         Item_Load 'Load Item on Name change
7
8 End Sub
9
10 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
11
12 End Sub
```

1 Option Explicit

2

1 Option Explicit

2

1	Option Explicit
2	


```

1 2 3 4
56 'Purchase Description
Range("H" & Target.Row).Value = ItemsDB.Range("I" & ItemRow).Value
'Purchase Cost
57 End Select
58 Range("G" & Target.Row).Value = 1 ' Set Default Qty to 1
59 Else 'Cleared Item
60 Range("F" & Target.Row & ":H" & Target.Row).ClearContents
61 If Range("J" & Target.Row).Value <> Empty Then
62 Dim ItemDbRow As Long
63 ItemDbRow = Range("J" & Target.Row).Value
64 OrderItemsDB.Range(ItemDbRow & ":" & ItemDbRow).EntireRow.Delete
65 Range("J" & Target.Row).ClearContents 'Clear Item DB Row
66 End If
67 End If
68 End If
69
70 'On Name Change, Set Address
71 If Not Intersect(Target, Range("F5" )) Is Nothing And Range("F5" ).Value <> Empty
And Range("B8" ).Value <> Empty Then
72 NameRow = Range("B8" ).Value 'Found Name Row
73 Range("F6" ).Value = NamesDB.Range("D" & NameRow).Value 'Address
74 Range("F7" ).Value = NamesDB.Range("E" & NameRow).Value & " " & _
75 NamesDB.Range("F" & NameRow).Value & " ," & _
76 NamesDB.Range("G" & NameRow).Value 'City, State & Zip
77 End If
78
79 'On Search Order Change
80 If Not Intersect(Target, Range("D8" )) Is Nothing And Range("D8" ).Value <> Empty
Then
81 If Range("B4" ).Value = Empty Then
82 MsgBox "Please make sure to enter a correct Order ID"
83 Range("D8" ).ClearContents
84 Range("D8" ).Select
85 Exit Sub
86 End If
87 Range("B5" ).Value = Range("D8" ).Value 'Set Order
88 Order_Load
89 End If
90
91 'On change of Payment Terms or Order Date (but not on order load)
92 If Not Intersect(Target, Range("I4,I6" )) Is Nothing And Range("B3" ).Value = False
Then
93 If Range("I4" ).Value = Empty Or Range("I6" ).Value = Empty Then Exit Sub
94 Dim TermRow As Long
95 On Error Resume Next
96 TermRow = Admin.Range("Terms" ).Find(Range("I6" ).Value, , xlValues, xlWhole).Row
97
98 On Error GoTo 0
99 If TermRow <> 0 Then Range("I7" ).Value = Range("I4" ).Value + Admin.Range("M" &
TermRow).Value 'Add Term Days To Order Date
100 End If
101
102 End Sub
103
104 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
105 If Shapes("AddNewGrp" ).Visible = True Then Shapes("AddNewGrp" ).Visible = msoFalse
106 If Shapes("CopyToGrp" ).Visible = True Then Shapes("CopyToGrp" ).Visible = msoFalse
107 End Sub

```

1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F5" )) Is Nothing Then Trans_LoadList 'Load
        transactions on anyfilter change
5 End Sub
6
7 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
8     If Target.CountLarge > 1 Then Exit Sub
9     If Not Intersect(Target, Range("D7:G9999" )) Is Nothing And Range("C" & Target.Row)
        .Value <> Empty Then
10         Range("B6" ).Value = Target.Row 'Set Selected Row
11         Range("B3" ).Value = Range("C" & Target.Row).Value 'Set Trans. ID
12         Trans_Load 'Run Macro to load transaction
13     End If
14 End Sub
```

1 Option Explicit

2

```
1
2 Option Explicit
3
4 Private Sub CancelBtn_Click()
5     AddEditName.Hide
6 End Sub
7
8
9 Private Sub SaveBtn_Click()
10     Dim NameFld As Control
11     Dim NameRow As Long, NameCol As Long
12     If Me.Field1.Value = Empty Or Me.NameType.Value = Empty Then
13         MsgBox "Please make sure to add a Name and Type before saving"
14         Exit Sub
15     End If
16     With NamesDB
17         'Check For new or Existing Name
18         If ActiveSheet.Range("B8").Value = Empty Then 'New Name
19             NameRow = .Range("A99999").End(xlUp).Row + 1 'First avail Row
20             .Range("A" & NameRow).Value = ActiveSheet.Range("B9").Value 'Unique Name ID
21         Else 'Existing Name
22             NameRow = ActiveSheet.Range("B8").Value 'Existing Name Row
23         End If
24         .Range("B" & NameRow).Value = Me.NameType.Value 'Set Name type
25         .Range("J" & NameRow).Value = Me.Status.Value 'Set Status
26         For NameCol = 3 To 9
27             Set NameFld = Me.Controls("Field" & NameCol - 2)
28             .Cells(NameRow, NameCol).Value = NameFld.Value 'Adds/Updates Field Date in db
29         Next NameCol
30         AddEditName.Hide
31         Names_UpdateAndSort 'Run Macro to update & sort names
32         If ActiveSheet.CodeName = "Orders" Then 'On Order Sheet
33             Orders.Range("F5").Value = Me.Field1.Value ' Set Name on Orders
34         Else 'On Transaction Sheet
35             Transactions.Range("J6").Value = Me.Field1.Value 'Set Name on Transactions
36         End If
37     End With
38 End Sub
```

```

1 Option Explicit
2
3 Sub BrowseForPicFolder()
4     Dim PicFolder As FileDialog
5     Set PicFolder = Application.FileDialog(msoFileDialogFolderPicker)
6     With PicFolder
7         .Title = "Browse for item picture folder"
8         .AllowMultiSelect = False
9         If .Show <> -1 Then GoTo NoSelection
10        Admin.Range("G24" ).Value = .SelectedItems(1)
11    End With
12    NoSelection:
13 End Sub
14
15 Sub BuildListOfAccounts()
16     Dim LastRow As Long, LastIncRow As Long, LastExpRow As Long
17     With Transactions
18         Application.ScreenUpdating = False
19         .Range("A14:A999" ).ClearContents 'Clear Previous Accounts
20         LastIncRow = Admin.Range("B999" ).End(xlUp).Row 'Last Income Account Row
21         LastExpRow = Admin.Range("C999" ).End(xlUp).Row 'Last Expense Account Row
22         If LastIncRow > 17 Then .Range("A14:A" & LastIncRow - 4).Value = Admin.Range(
23             "B18:B" & LastIncRow).Value 'Bring Over Income Accounts
24         If LastExpRow > 17 Then .Range("A" & LastIncRow - 3 & ":A" & LastIncRow - 3 +
25             LastExpRow - 18).Value = Admin.Range("C18:C" & LastExpRow).Value 'Bring Over
26             Expense Accounts
27         LastRow = .Range("A999" ).End(xlUp).Row 'Last Account Row
28         'Sort Alphabetically
29         With .Sort
30             .SortFields.Clear
31             .SortFields.Add Key:=Transactions.Range("A14" ), SortOn:=xlSortOnValues, Order
32             :=xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
33             .SetRange Transactions.Range("A14:A" & LastRow)
34             .Apply
35         End With
36         Application.ScreenUpdating = True
37     End With
38 End Sub
39
40 Sub Menu_GoToSheet()
41     Sheets(Application.Caller).Activate
42 End Sub

```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, LastTransResultRow As Long, ResultRow As Long
3 Dim TopPos As Double, LeftPos As Double
4 Dim PicFolder As String, PicPath As String
5 Dim PicShp As Shape
6 Sub Dashboard_Refresh()
7
8     'Clear Any Pictures shapes
9     For Each PicShp In Dashboard.Shapes
10         On Error Resume Next
11         If InStr(PicShp.Name, "ItemPic" ) > 0 Then PicShp.Delete
12         On Error GoTo 0
13     Next PicShp
14
15     'Get Order Data For Profit & Loss
16     With OrdersDB
17         .Range("AA3:AE9999" ).ClearContents 'Clear Existing Data
18         LastRow = .Range("A99999" ).End(xlUp).Row
19         If LastRow < 4 Then GoTo NoOrders
20         .Range("M4:M" & LastRow).Formula = .Range("M2" ).Formula 'Copy Down Balance
21         .Range("A3:L" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("S2:U4" ), CopyToRange:=.Range("AA2:AD2" ), Unique:=True
22         LastResultRow = .Range("AA99999" ).End(xlUp).Row
23         If LastResultRow < 3 Then GoTo NoOrders
24
25         NoOrders:
26         If LastResultRow < 3 Then LastResultRow = 2 'Set Min. Last Result Row
27     End With
28
29     'Get Transaction Data For Profit & Loss
30     With TransDB
31         LastRow = .Range("A99999" ).End(xlUp).Row
32         If LastRow < 4 Then GoTo NoTrans
33         .Range("A3:F" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("V2:W3" ), CopyToRange:=.Range("AA2:AD2" ), Unique:=True
34         LastTransResultRow = .Range("AA99999" ).End(xlUp).Row
35         If LastTransResultRow < 3 Then GoTo NoTrans
36
37
38         If LastResultRow < 3 Then LastResultRow = 2 'Set Min. Last Result Row
39
40     'Combine Transactions into a single table
41     OrdersDB.Range("AA" & LastResultRow + 1 & ":AD" & LastResultRow + 1 +
42     LastTransResultRow - 3).Value = .Range("AA3:AD" & LastTransResultRow).Value
43     NoTrans:
44     LastRow = OrdersDB.Range("AA99999" ).End(xlUp).Row
45     OrdersDB.Range("AE3:AE" & LastRow).Formula = OrdersDB.Range("AE1" ).Formula
46     'Bring Down Account Type Formula
47 End With
48
49     'Get Top 10 Needed Items
50     With ItemsDB
51         .Range("T3:X99999" ).ClearContents 'Clear Previous results
52         LastRow = .Range("A99999" ).End(xlUp).Row
53         If LastRow < 4 Then GoTo NoItems
54         .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("Q2:R3" ), CopyToRange:=.Range("T2:V2" ), Unique:=True
55         LastResultRow = .Range("T99999" ).End(xlUp).Row
56         If LastResultRow < 3 Then GoTo NoItems

```

```

1 2
55 .Range("W3:X" & LastResultRow).Formula = .Range("W1:X1" ).Formula 'Copy Down
Qty On Hand & Qty Difference Formulas
56 If LastResultRow < 4 Then GoTo NoSort: 'No sorting on a single Item
57
58 With .Sort
59 .SortFields.Clear
60 .SortFields.Add Key:=ItemsDB.Range("X3" ), Order:=xlDescending
61 .SetRange ItemsDB.Range("T3:X" & LastResultRow)
62 .Apply
63 End With
64 If LastResultRow > 12 Then LastResultRow = 12 'Set 10 Maximum
65 With .Sort
66 .SortFields.Clear
67 .SortFields.Add Key:=ItemsDB.Range("X3" ), Order:=xlAscending
68 .SetRange ItemsDB.Range("T3:X" & LastResultRow)
69 .Apply
70 End With
71 NoSort:
72 'Add Pictures (if Available)
73 PicFolder = [ItemPicFolder]
74 If PicFolder = "" Or Dir(PicFolder, vbDirectory) = "" Then Exit Sub
75 LeftPos = Dashboard.Range("N6" ).Left + 48 'Set initial left positions
76 TopPos = Dashboard.Range("N6" ).Top - 4 'Set initial top position
77 For ResultRow = LastResultRow To 3 Step -1
78 PicPath = PicFolder & "\" & .Range("T" & ResultRow).Value 'Picture Path
79 Dashboard.Pictures.Insert(PicPath).Name = "ItemPic" & ResultRow
80 With Dashboard.Shapes("ItemPic" & ResultRow)
81 .Top = TopPos
82 .Left = LeftPos + ((26 - Len(ItemsDB.Range("V" & ResultRow).Value))) * 4.5
83 .Height = 25
84 .Width = 25
85 End With
86 TopPos = TopPos + 31 'Increment Top Position
87
88 Next ResultRow
89 NoItems:
90 End With
91
92 'Get Sales Data based on date Range
93 With OrdersDB
94 .Range("AK3:AL9999" ).ClearContents 'Clear Existing Data
95 LastRow = .Range("A9999" ).End(xlUp).Row
96 If LastRow < 4 Then GoTo NoOrders
97 .Range("A3:L" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"S2:U3" ), CopyToRange:=.Range("AK2:AL2" ), Unique:=True
98
99 'Get Outstanding Invoice Data
100 .Range("A3:M" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"AQ2:AS3" ), CopyToRange:=.Range("AU2:AX2" ), Unique:=True
101
102 'Get Outstanding Purchase Order Data
103 .Range("A3:M" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"BA2:BC3" ), CopyToRange:=.Range("BE2:BH2" ), Unique:=True
104 End With
105 ThisWorkbook.RefreshAll 'Refresh All Data
106 End Sub

```

```

1 Option Explicit
2 Dim ItemRow As Long, ItemCol As Long, LastRow As Long, LastResultRow As Long, ResultRow
  As Long
3 Dim PicFolder As String, PicPath As String
4 Dim ItemPic As FileDialog
5
6 Sub Item_BrowseItemPic()
7     On Error Resume Next
8     Items.Shapes("ItemPic" ).Delete 'Remove picture if existing
9     On Error GoTo 0
10    PicFolder = [ItemPicFolder] & "/" 'Get Item Picture folder
11    'Check for Mobile Data Folder
12    If PicFolder = "/" Or Dir(PicFolder, vbDirectory) = "" Then
13        MsgBox "Please set a folder for the item Pictures"
14        BrowseForPicFolder 'Run Macro to browse for Picture folder
15        PicFolder = [ItemPicFolder] & "/"
16        If PicFolder = "/" Or Dir(PicFolder, vbDirectory) = "" Then Exit Sub 'Exit on
  Empty
17    End If
18    Set ItemPic = Application.FileDialog(msoFileDialogFilePicker)
19    With ItemPic
20        .Title = "Please select an Item Picture"
21        .Filters.Add "Picture Files", "*.jpg,*.png,*.gif, *.jpeg *.GIF", 1
22        .AllowMultiSelect = False
23        If .Show <> -1 Then GoTo NoSelection
24        If .SelectedItems(1) = PicFolder & Dir(.SelectedItems(1)) Then GoTo SkipCopy
25        If Dir(PicFolder & Dir(.SelectedItems(1)), vbDirectory) <> "" Then Kill (
  PicFolder & Dir(.SelectedItems(1))) 'Delete if existing
26        FileCopy .SelectedItems(1), PicFolder & Dir(.SelectedItems(1)) 'Copy Picture
  into Item folder
27        SkipCopy:
28        Items.Range("B7" ).Value = Dir(.SelectedItems(1)) 'Place file name
29    End With
30    Item_ShowItemPic 'Run Macro to Display Item Picture
31    NoSelection:
32 End Sub
33
34 Sub Item_ClearItemPic()
35     On Error Resume Next
36     Items.Shapes("ItemPic" ).Delete 'Remove picture if existing
37     On Error GoTo 0
38     Items.Range("B7" ).ClearContents
39 End Sub
40
41 Sub Item_Delete()
42     If MsgBox("Are you sure you want to delete this Item?" , vbYesNo, "Delete Item" ) =
  vbNo Then Exit Sub
43     With Items
44         If .Range("B4" ).Value = Empty Then GoTo NotSaved
45         ItemRow = .Range("B4" ).Value 'Item Row
46         ItemsDB.Range(ItemRow & ":" & ItemRow).EntireRow.Delete
47         NotSaved:
48         Item_New 'Clear Out All Form Fields
49         Items_UpdateSortedNames 'Reload Item list
50     End With
51 End Sub
52
53
54
55 Sub Item_Load()
56     With Items

```



```

1 2
57 .Range("B3" ).Value = .Range("B6" ).Value 'Set Item ID
58 .Range("E3,E5,H5,K5,F7:H7,F9:H9,F11:H11,K7,K9" ).ClearContents
59 On Error Resume Next
60 .Shapes("ItemPic" ).Delete 'Remove picture if existing
61 On Error GoTo 0
62 If .Range("B4" ).Value = Empty Then
63     MsgBox "Please make sure to select a correct Item"
64     Exit Sub
65 End If
66 Application.ScreenUpdating = False
67 ItemRow = .Range("B4" ).Value 'Item Row
68 'Load Item details
69 For ItemCol = 2 To 10
70     .Range(ItemsDB.Cells(1, ItemCol).Value).Value = ItemsDB.Cells(ItemRow, ItemCol
71     ).Value 'Save/Update Items DB
72 Next ItemCol
73 If .Range("B7" ).Value <> "" Then Item_ShowItemPic
74 End With
75 Application.ScreenUpdating = True
76 End Sub

77 Sub Item_New()
78     With Items
79         .Range("B3,E3,E5,H5,K5,F7:H7,F9:H9,F11:H11,K7,K9" ).ClearContents
80         On Error Resume Next
81         .Shapes("ItemPic" ).Delete
82         On Error GoTo 0
83         .Range("E5" ).Select
84     End With
85 End Sub

86 Sub Item_SavedMsg()
87     With Items.Shapes("ItemSavedMsg" )
88         Dim i As Long, Delay As Double, StartTime As Double
89         .Visible = msoCTrue
90         For i = 1 To 100
91             .Fill.Transparency = i / 100
92             Delay = 0#
93             StartTime = Timer
94             Do
95                 DoEvents
96             Loop While Timer - StartTime < Delay
97         Next i
98         .Visible = msoFalse
99     End With
100 End Sub

101
102 Sub Item_SaveUpdate()
103     With Items
104         If .Range("E5" ).Value = Empty Or .Range("H5" ).Value = Empty Then
105             MsgBox "Please make sure to add an Item Name & Type before saving this Item"
106             Exit Sub
107         End If
108
109         If .Range("B4" ).Value = Empty Then 'New Item
110             ItemRow = ItemsDB.Range("A99999" ).End(xlUp).Row + 1 'New Database Row
111             .Range("B3" ).Value = .Range("B5" ).Value 'Set Next Item ID
112             ItemsDB.Range("A" & ItemRow).Value = .Range("B3" ).Value 'Set Item ID
113         Else 'Existing Item
114             ItemRow = .Range("B4" ).Value 'Existing Item Row
115         End If
116         'Add/Update Data on Local Workbook

```

```

1 2
117 For ItemCol = 2 To 10
118     ItemsDB.Cells(ItemRow, ItemCol).Value = .Range(ItemsDB.Cells(1, ItemCol).Value
        ).Value 'Save/Update Items DB
119 Next ItemCol
120 Items_UpdateSortedNames 'Run macro to update names list
121 Item_SavedMsg 'Run Fade Out Message
122 End With
123 End Sub
124 Sub Item_ShowItemPic()
125     With Items
126         On Error Resume Next
127         .Shapes("ItemPic").Delete 'Remove picture if existing
128         On Error GoTo 0
129
130         'Check for Picture Folder
131         PicFolder = [ItemPicFolder] & "/" 'Get Item Picture folder
132         'Check for Mobile Data Folder
133         If PicFolder = "/" Or Dir(PicFolder, vbDirectory) = "" Then
134             MsgBox "Please set a folder for the item Pictures"
135             BrowseForPicFolder 'Run Macro to browse for Picture folder
136             PicFolder = [ItemPicFolder] & "/"
137             If PicFolder = "/" Or Dir(PicFolder, vbDirectory) = "" Then Exit Sub 'Exit
                on Empty
138         End If
139         PicPath = PicFolder & .Range("B7").Value 'Picture File Path
140
141         If Dir(PicPath, vbDirectory) <> "" Then 'Check For Accurate Path
142             .Pictures.Insert(PicPath).Name = "ItemPic"
143             With .Shapes("ItemPic")
144                 .LockAspectRatio = msoCTrue
145                 If .Width > .Height Then .Width = 90 Else .Height = 80 'Set Width / Height
146
147                 .Left = Items.Range("M4").Left + (Items.Range("M1").Width - .Width) / 2
                    'Center Pic Horizontally
148                 .Top = Items.Range("M4").Top + (Items.Range("4:10").Height - .Height) / 2
                    'Center Pic Vertically
149                 .Visible = msoCTrue
150             End With
151         End If
152     End With
153 End Sub
154 Sub Items_UpdateSortedNames()
155     With ItemsDB
156         'Update Sorted Names List
157         LastRow = .Range("A99999").End(xlUp).Row
158         .Range("N3:N99999").ClearContents 'Clear Existing Items
159         .Range("N3:N" & LastRow - 1).Value = .Range("B4:B" & LastRow).Value 'Bring
            over names
160     With .Sort
161         .SortFields.Clear
162         .SortFields.Add Key:=ItemsDB.Range("N3"), SortOn:=xlSortOnValues, Order:=
            xlAscending, DataOption:=xlSortNormal 'Sort By Due Date
163         .SetRange ItemsDB.Range("N3:N" & LastRow)
164         .Apply
165     End With
166 End With
167 End Sub

```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long
3 Sub Names_UpdateAndSort()
4     With NamesDB
5         .Range("S3:V99999").ClearContents 'Clear Prior Results
6         LastRow = .Range("A99999").End(xlUp).Row
7         If LastRow < 4 Then Exit Sub
8         Application.ScreenUpdating = False
9         .Range("S3:S" & LastRow - 1).Value = .Range("C4:C" & LastRow).Value 'Get All
        Names List
10        'Sort Alphabetically
11        With .Sort
12            .SortFields.Clear
13            .SortFields.Add Key:=NamesDB.Range("S3" ), SortOn:=xlSortOnValues, Order:=
                xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
14            .SetRange NamesDB.Range("S3:S" & LastRow)
15            .Apply
16        End With
17        'Get Unique Customer List
18        .Range("B3:C" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
                "O2:O3" ), CopyToRange:=.Range("T2" ), Unique:=True
19        LastResultRow = .Range("T99999").End(xlUp).Row
20        If LastResultRow < 4 Then GoTo NoCust 'Exit on 1 or no results
21        'Sort Alphabetically
22        With .Sort
23            .SortFields.Clear
24            .SortFields.Add Key:=NamesDB.Range("T3" ), SortOn:=xlSortOnValues, Order:=
                xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
25            .SetRange NamesDB.Range("T3:T" & LastResultRow)
26            .Apply
27        End With
28        NoCust:
29        'Get Unique Vendor List
30        .Range("B3:C" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
                "P2:P3" ), CopyToRange:=.Range("U2" ), Unique:=True
31        LastResultRow = .Range("U99999").End(xlUp).Row
32        If LastResultRow < 4 Then GoTo NoVend 'Exit on 1 or no results
33        'Sort Alphabetically
34        With .Sort
35            .SortFields.Clear
36            .SortFields.Add Key:=NamesDB.Range("U3" ), SortOn:=xlSortOnValues, Order:=
                xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
37            .SetRange NamesDB.Range("U3:U" & LastResultRow)
38            .Apply
39        End With
40        NoVend:
41
42        'Get Unique Employee List
43        .Range("B3:C" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
                "Q2:Q3" ), CopyToRange:=.Range("V2" ), Unique:=True
44        LastResultRow = .Range("V99999").End(xlUp).Row
45        If LastResultRow < 4 Then GoTo NoEmpl 'Exit on 1 or no results
46        'Sort Alphabetically
47        With .Sort
48            .SortFields.Clear
49            .SortFields.Add Key:=NamesDB.Range("V3" ), SortOn:=xlSortOnValues, Order:=
                xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
50            .SetRange NamesDB.Range("V3:V" & LastResultRow)
51            .Apply
52        End With
53        NoEmpl:

```

```
54 | 1 2 Application.ScreenUpdating = True
55 | End With
56 | End Sub
```

```

1 Option Explicit
2 Dim OrdRow As Long, OrdCol As Long, LastItemRow As Long, ItemRow As Long, ResultRow As Long, LastResultRow As Long, OrdDBRow As Long
3 Dim TypeNumb As Long, OrdNumb As Long, OrdID As Long, OrderItemRow As Long, OrderItemDBRow As Long, LastOrderItemDBRow As Long, IDRow As Long
4 Dim DefStatusRow As Long, OrderTypeNumb As Long, DefFootMsgRow As Long, TermRow As Long, TypeRow As Long
5 Dim OrderType As String, StatusType As String, FileName As String, OrdStatus As String, OrdAcct As String
6 Dim OrdDate As Date
7 Dim OutApp As Object, OutMail As Object
8
9 Sub Order_CopyTo()
10     With Orders
11         OrderType = .Shapes(Application.Caller).TextFrame2.TextRange.Text
12         StatusType = .Shapes(Application.Caller).Name 'Set Status Type Name
13         .Range("B2").Value = OrderType 'Order Type
14         OrderTypeNumb = .Range("B1").Value 'Set Order Type Number
15         .Range("B3").Value = True 'Set order load to true
16         .Range("B5,D8:D12,J10:J34").ClearContents 'Clear Order Fields
17         .Range("I4").Value = Date 'Set current date
18         .Range("B5").Value = .Range("B7").Value 'Set Next Order ID
19         .Range("B3").Value = False 'Set order load to false
20         .Shapes("CopyToGrp").Visible = msoFalse 'hide add new group
21
22
23         .Range("D10,D11,F5").Validation.Delete 'Delete Status and Vendor/Customer Validation & Account Validation
24         .Range("D10").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:="=" & "Status_" & StatusType 'Set Status Type
25
26         Select Case OrderType
27             Case Is = [EstimOrder] 'Estimate
28                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:="=Customer_Name" 'Set Customer Validation
29             Case Is = [WrkOrder] 'Work Order
30                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:="=Customer_Name" 'Set Customer Validation
31             Case Is = [InvOrder] 'Invoice
32                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:="=Customer_Name" 'Set Customer Validation
33                 .Range("D11").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:="=Accounts_Income" 'Set Income Accounts Validation
34                 .Range("D11").Value = [InvIncAcct] 'Set Default Income Account
35             Case Is = [PurchOrder] 'Purchase Order
36                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:="=Vendor_Name" 'Set Vendor Validation
37                 .Range("D11").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:="=Accounts_Expenses" 'Set Expense Accounts Validation
38                 .Range("D11").Value = [PurchExpAcct] 'Set Default Expense Account
39         End Select
40
41         'Set Default Payment Terms on Invoices & Purchase Orders
42         If OrderType = [InvOrder] Or OrderType = [PurchOrder] Then
43             On Error Resume Next
44             TermRow = Admin.Range("N15:N21").Find(Chr(252), , xlValues, xlWhole).Row
45             On Error GoTo 0
46             If TermRow <> 0 Then .Range("I6").Value = Admin.Range("L" & TermRow).Value 'Set Default Term
47         End If
48

```

```

1 2
49 'Set Default Status Based on Order Type
50 On Error Resume Next
51 DefStatusRow = Admin.Range("StatusDef_" & StatusType).Find(Chr(252), , xlValues,
xlWhole).Row 'Default Status Row
52 On Error GoTo 0
53 If DefStatusRow <> 0 Then .Range("D10" ).Value = Admin.Cells(DefStatusRow, Admin.
Range("Status_" & StatusType).Column).Value 'Set Default Status
54 'Set Default Footer Message Based On Order Type
55 On Error Resume Next
56 DefFootMsgRow = Range(Admin.Cells(28, OrderTypeNumb + 17), Admin.Cells(39,
OrderTypeNumb + 17)).Find(Chr(252), , xlValues, xlWhole).Row
57 On Error GoTo 0
58 If DefFootMsgRow <> 0 Then .Range("D9" ).Value = Admin.Range("F" & DefFootMsgRow
).Value 'Set Default Footer Message
59 .Range("F5" ).Select
60 End With
61 End Sub

62
63 Sub Order_Delete()
64 If MsgBox("Are you sure you want to delete this order?" , vbYesNo, "Delete Order" )
= vbNo Then Exit Sub
65 If Orders.Range("B6" ).Value = "" Then GoTo NotSaved 'Order Not yet saved
66 OrdRow = Orders.Range("B6" ).Value 'Order Row
67 OrdersDB.Range(OrdRow & ":" & OrdRow).EntireRow.Delete 'Delete Order DB Row
68 OrderType = Orders.Range("B2" ).Value 'Set Order Type
69 With OrderItemsDB
70 LastOrderItemDBRow = .Range("A99999" ).End(xlUp).Row
71 .Range("A3:L" & LastOrderItemDBRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
.Range("N2:N3" ), CopyToRange:=.Range("R2:W2" ), Unique:=True
72 LastResultRow = .Range("R99999" ).End(xlUp).Row 'Last Results Row
73 If LastResultRow < 3 Then GoTo NoDBItems
74 If LastResultRow < 4 Then GoTo NoSort
75 With .Sort
76 .SortFields.Clear
77 .SortFields.Add Key:=OrderItemsDB.Range("W3" ), Order:=xlDescending
78 .SetRange OrderItemsDB.Range("R3:W" & LastResultRow)
79 .Apply
80 End With
81 NoSort:
82 For ResultRow = 3 To LastResultRow
83 OrderItemDBRow = .Range("W" & ResultRow).Value 'Order Item Row
84 .Range(OrderItemDBRow & ":" & OrderItemDBRow).EntireRow.Delete
85 Next ResultRow
86 NoDBItems:
87 NotSaved:
88 MsgBox "This " & OrderType & " has been deleted"
89 Order_New
90 End With
91 End Sub

92
93
94
95 Sub Order_Email()
96 Dim Subj As String, Mesg As String
97 Dim VarRow As Long
98 With Orders
99 .PageSetup.PrintArea = "$E$3:$I$39" 'Set Print Area
100 FileName = ThisWorkbook.Path & "\" & .Range("B2" ).Value & "_" & .Range("B5" ).
Value & ".pdf"
101 If Dir(FileName, vbDirectory) <> "" Then Kill (FileName) 'Delete file if exists

```

```

1 2
102 .ExportAsFixedFormat xlTypePDF, FileName, , , False
103 Set OutApp = CreateObject("Outlook.Application")
104 Set OutMail = OutApp.CreateItem(0)
105 Subj = .Range("AI4").Value 'Subject
106 Mesg = .Range("AI5").Value 'Email Message
107 For VarRow = 5 To 9
108     Subj = Replace(Subj, .Range("AJ" & VarRow).Value, Format(.Range("AK" &
109     VarRow).Value, .Range("AK" & VarRow).NumberFormat))
109     Mesg = Replace(Mesg, .Range("AJ" & VarRow).Value, Format(.Range("AK" &
110     VarRow).Value, .Range("AK" & VarRow).NumberFormat))
110 Next VarRow
111 With OutMail
112     .To = Orders.Range("B10").Value 'Customer /Vendor /Employee email
113     .Attachments.Add FileName
114     .Subject = Subj
115     .Body = Mesg
116     .Display 'To send use .Send
117 End With
118 On Error GoTo 0
119 Set OutMail = Nothing
120 End With
121 End Sub

122
123
124
125 Sub Order_Load()
126     With Orders
127         If .Range("B6").Value = Empty Then
128             MsgBox "Please enter a correct order #"
129             Exit Sub
130         End If
131         OrdRow = .Range("B6").Value 'Order Row
132         .Range("B3").Value = True 'Set order load to true
133         .Range("D8:D12,F5:F7,I4:I7,E10:H34,J10:J34").ClearContents 'Clear Order Fields
134         Application.ScreenUpdating = False
135         For OrdCol = 2 To 11
136             .Range(OrdersDB.Cells(1, OrdCol).Value).Value = OrdersDB.Cells(OrdRow, OrdCol)
137             .Value 'Bring in Order data
138         Next OrdCol
139         OrderType = .Range("B2").Value 'Set Order Type
140         .Range("D10,D11,F5").Validation.Delete 'Delete Status and Vendor/Customer
141         Validation & Account Validation
142         Select Case OrderType
143             Case Is = [EstimOrder] 'Estimate
144                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
145                 := "Customer_Name" 'Set Customer Validation
146                 StatusType = "Status_Estimate"
147             Case Is = [WrkOrder] 'Work Order
148                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
149                 := "Customer_Name" 'Set Customer Validation
150                 StatusType = "Status_Work_Order"
151             Case Is = [InvOrder] 'Invoice
152                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
153                 := "Customer_Name" 'Set Customer Validation
154                 .Range("D11").Validation.Add xlValidateList, xlValidAlertInformation,
155                 Formula1 := "Accounts_Income" 'Set Income Accounts Validation
156                 StatusType = "Status_Invoice"
157             Case Is = [PurchOrder] 'Purchase Order
158                 .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
159                 := "Vendor_Name" 'Set Vendor Validation

```

1 2 3


```

154     .Range("D11" ).Validation.Add xlValidateList, xlValidAlertInformation,
        Formula1:="=Accounts_Expenses" 'Set Expense Accounts Validation
155     StatusType = "Status_Purchase_Order"
156 End Select
157     .Range("D10" ).Validation.Add xlValidateList, xlValidAlertInformation, Formula1:=
        "=" & StatusType 'Set Status Type
158
159     'Load Order Items
160     LastOrderItemDBRow = OrderItemsDB.Range("A99999" ).End(xlUp).Row 'Last Item Row
161
162     If LastOrderItemDBRow < 4 Then GoTo NoDBItems
163     OrderItemsDB.Range("A3:L" & LastOrderItemDBRow).AdvancedFilter xlFilterCopy,
        CriteriaRange:=OrderItemsDB.Range("N2:N3" ), CopyToRange:=OrderItemsDB.Range(
        "R2:W2" ), Unique:=True
164     LastResultRow = OrderItemsDB.Range("R99999" ).End(xlUp).Row 'Last Results Row
165     If LastResultRow < 3 Then GoTo NoDBItems
166     For ResultRow = 3 To LastResultRow
167         ItemRow = OrderItemsDB.Range("V" & ResultRow).Value 'Order Item Row
168         .Range("E" & ItemRow & ":H" & ItemRow).Value = OrderItemsDB.Range("R" &
            ResultRow & ":U" & ResultRow).Value 'Item Name, Desc. Qty, Amount
169         .Range("J" & ItemRow).Value = OrderItemsDB.Range("W" & ResultRow).Value
            'Items DB Row
170     Next ResultRow
171     NoDBItems:
172     .Range("B3" ).Value = False 'Set order load to False
173     Application.ScreenUpdating = True
174 End With
175 End Sub
176
177 Sub Order_New()
178     With Orders
179         If ActiveSheet.Name = "Home" Then 'Access From Home Screen
180             OrderType = Replace(Home.Shapes(Application.Caller).Name, "_", " ")
181             On Error Resume Next
182             TypeRow = Admin.Range("B5:B8" ).Find(OrderType, , xlValues, xlWhole).Row
183             On Error GoTo 0
184             If TypeRow = 0 Then Exit Sub
185             OrderType = Admin.Range("C" & TypeRow).Value
186             StatusType = Home.Shapes(Application.Caller).Name
187             Orders.Activate
188         Else 'From Order Screen
189             OrderType = .Shapes(Application.Caller).TextFrame2.TextRange.Text
190             StatusType = .Shapes(Application.Caller).Name 'Set Status Type Name
191             If OrderType = "Delete Order" Then 'On Add New From Delete Order Macro
192                 OrderType = [InvOrder] 'Set Default To Invoice
193                 StatusType = "Invoice"
194             End If
195         End If
196     End With
197     .Range("B2" ).Value = OrderType 'Order Type
198     OrderTypeNumb = .Range("B1" ).Value 'Set Order Type Number
199     .Range("B3" ).Value = True 'Set order load to true
200     .Range("B5,D8:D12,F5:F7,I4:I7,E10:H34,J10:J34" ).ClearContents 'Clear Order
        Fields
201     .Range("I4" ).Value = Date 'Set current date
202     .Range("B5" ).Value = .Range("B7" ).Value 'Set Next Order ID
203
204     .Shapes("AddNewGrp" ).Visible = msoFalse 'hide add new group
205     .Range("D10,D11,F5" ).Validation.Delete 'Delete Status and Vendor/Customer
        Validation & Account Validation
        On Error Resume Next

```



```

1 2
206 .Range("D10").Validation.Add xlValidateList, xlValidAlertInformation, Formula1:=
207     "=" & "Status_" & StatusType 'Set Status Type
208     On Error GoTo 0
209     Select Case OrderType
210     Case Is = [EstimOrder] 'Estimate
211         .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
212             := "Customer_Name" 'Set Customer Validation
213     Case Is = [WrkOrder] 'Work Order
214         .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
215             := "Customer_Name" 'Set Customer Validation
216     Case Is = [InvOrder] 'Invoice
217         .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
218             := "Customer_Name" 'Set Customer Validation
219         .Range("D11").Validation.Add xlValidateList, xlValidAlertInformation,
220             Formula1 := "Accounts_Income" 'Set Income Accounts Validation
221         .Range("D11").Value = [InvIncAcct] 'Set Default Income Account
222     Case Is = [PurchOrder] 'Purchase Order
223         .Range("F5").Validation.Add xlValidateList, xlValidAlertInformation, Formula1
224             := "Vendor_Name" 'Set Vendor Validation
225         .Range("D11").Validation.Add xlValidateList, xlValidAlertInformation,
226             Formula1 := "Accounts_Expenses" 'Set Expense Accounts Validation
227         .Range("D11").Value = [PurchExpAcct] 'Set Default Expense Account
228     End Select
229     'Set Default Payment Terms on Invoices & Purchase Orders
230     If OrderType = [InvOrder] Or OrderType = [PurchOrder] Then
231         On Error Resume Next
232         TermRow = Admin.Range("N15:N21").Find(Chr(252), , xlValues, xlWhole).Row
233         On Error GoTo 0
234         If TermRow <> 0 Then .Range("I6").Value = Admin.Range("L" & TermRow).Value
235             'Set Default Term
236     End If
237     'Set Default Status Based on Order Type
238     On Error Resume Next
239     DefStatusRow = Admin.Range("StatusDef_" & StatusType).Find(Chr(252), , xlValues,
240         xlWhole).Row 'Default Status Row
241     On Error GoTo 0
242     If DefStatusRow <> 0 Then .Range("D10").Value = Admin.Cells(DefStatusRow, Admin.
243         Range("Status_" & StatusType).Column).Value 'Set Default Status
244     'Set Default Footer Message Based On Order Type
245     On Error Resume Next
246     DefFootMsgRow = Range(Admin.Cells(28, OrderTypeNum + 17), Admin.Cells(39,
247         OrderTypeNum + 17)).Find(Chr(252), , xlValues, xlWhole).Row
248     On Error GoTo 0
249     If DefFootMsgRow <> 0 Then .Range("D9").Value = Admin.Range("F" & DefFootMsgRow
250         ).Value 'Set Default Footer Message
251     .Range("B3").Value = False 'Set order load to false
252     .Range("F5").Select
253 End With
254 End Sub

255 Sub Order_Print()
256     Orders.PageSetup.PrintArea = "$E$3:$I$39" 'Set Print Area
257     Orders.PrintOut , , , True, , , False
258 End Sub

259 Sub Order_SavedMsg()

```

```

1
255 With Orders.Shapes("OrderSavedMsg" )
256     Dim i As Long, Delay As Double, StartTime As Double
257     .Visible = msoCTrue
258     For i = 1 To 100
259         .Fill.Transparency = i / 100
260         Delay = 0.009
261         StartTime = Timer
262         Do
263             DoEvents
264             Loop While Timer - StartTime < Delay
265         Next i
266     .Visible = msoFalse
267 End With
268 End Sub
269 Sub Order_SaveUpdate()
270     With Orders
271         OrderType = .Range("B2" ).Value 'Set Order Type
272
273         'Check For Required Fields
274         If .Range("F5" ).Value = Empty Then 'Missing Customer/Vendor
275             MsgBox "Please be sure to add a " & .Range("E5" ).Value & " before saving
                this " & OrderType
                Exit Sub
276         End If
277
278         If .Range("I4" ).Value = Empty Then 'Missing date
279             MsgBox "Please be sure to add an order date before saving this " & OrderType
280             Exit Sub
281         End If
282
283         If .Range("B6" ).Value = Empty Then 'New Order
284             OrdRow = OrdersDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail. row
285             .Range("B5" ).Value = .Range("B7" ).Value 'Next Order Id
286             OrdersDB.Range("A" & OrdRow).Value = .Range("B5" ).Value 'Order ID
287         Else 'Existing Order
288             OrdRow = .Range("B6" ).Value 'existing Order Row
289         End If
290
291         OrdID = .Range("B5" ).Value 'Order ID
292         OrdDate = .Range("I4" ).Value 'Order Date
293         OrdStatus = .Range("D10" ).Value 'Order Status
294         OrdAcct = .Range("D11" ).Value 'Order Account
295         For OrdCol = 2 To 12
296             OrdersDB.Cells(OrdRow, OrdCol).Value = .Range(OrdersDB.Cells(1, OrdCol).Value)
                .Value 'Save Data to db
297         Next OrdCol
298
299         'Add/Update Order Items
300         LastItemRow = .Range("E34" ).End(xlUp).Row 'Last Item Row
301         If LastItemRow < 10 Then GoTo NoItems
302         For ItemRow = 10 To LastItemRow
303             If .Range("J" & ItemRow).Value <> Empty Then 'Existing DB Row
304                 OrderItemDBRow = .Range("J" & ItemRow).Value 'Order item Row
305             Else 'New DBRow
306                 OrderItemDBRow = OrderItemsDB.Range("A99999" ).End(xlUp).Row + 1 'First
                Avail
307                 OrderItemsDB.Range("A" & OrderItemDBRow).Value = OrdID 'Order ID
308                 OrderItemsDB.Range("K" & OrderItemDBRow).Value = ItemRow 'Order Item Row
309                 OrderItemsDB.Range("L" & OrderItemDBRow).Value = "=Row()" 'Database Row
310                 .Range("J" & ItemRow).Value = OrderItemDBRow 'Set Database Row
311             End If
312             OrderItemsDB.Range("B" & OrderItemDBRow).Value = OrderType 'Order Type
313             OrderItemsDB.Range("C" & OrderItemDBRow).Value = OrdDate 'Order Date

```

1 2 3

```
1 2 3
312 OrderItemsDB.Range("D" & OrderItemDBRow).Value = OrdStatus 'Order Status
313 OrderItemsDB.Range("E" & OrderItemDBRow).Value = OrdAcct 'Order Account
314 OrderItemsDB.Range("F" & OrderItemDBRow & ":J" & OrderItemDBRow).Value = .
Range("E" & ItemRow & ":I" & ItemRow).Value 'All Item Details
315 Next ItemRow
316 NoItems:
317 Order_SavedMsg 'Run fade out message
318 End With
319 End Sub
```

```

1 Option Explicit
2 Dim OrdRow As Long, OrdCol As Long, OrderItemDBRow As Long, LastItemRow As Long,
ItemRow As Long
3 Dim OrdNumb As Long, LastOrderItemDBRow As Long, LastResultRow As Long, ResultRow As
Long
4 Dim FoundItem As Range
5 Dim OrdType As String
6 Dim OrdShp As Shape
7
8 Sub Order_AddNewMenu()
9     'Toggle Order Type Drop Down List
10    If Orders.Shapes("AddNewGrp").Visible = True Then
11        Orders.Shapes("AddNewGrp").Visible = msoFalse
12    Else
13        Orders.Shapes("AddNewGrp").Visible = msoCTrue
14    End If
15 End Sub
16
17 Sub Order_CopyToMenu()
18     'Toggle Order Type Drop Down List
19    If Orders.Shapes("CopyToGrp").Visible = True Then
20        Orders.Shapes("CopyToGrp").Visible = msoFalse
21    Else
22        Orders.Shapes("CopyToGrp").Visible = msoCTrue
23    End If
24 End Sub
25
26
27
28 Sub Orders_AddEditName()
29     Dim NameFld As Control
30     Dim NameRow As Long, NameCol As Long
31     Unload AddEditName 'Clear Fields In Form
32     With Orders
33         If .Range("B8").Value = Empty Then 'New Name
34             AddEditName.Field1.Value = .Range("F5").Value
35             If .Range("B2").Value = [PurchaseOrder] Then 'Purchase Orders
36                 AddEditName.NameType.Value = [VendorNm] 'Set Default to Vendor
37             Else 'All Other Orders
38                 AddEditName.NameType.Value = [CustomerNm] 'Set Default to Customer
39             End If
40             AddEditName.Status.Value = "Active" 'Set Default to Active Status
41         Else 'Existing Name
42             NameRow = .Range("B8").Value 'Set Name Row
43             For NameCol = 3 To 9
44                 Set NameFld = AddEditName.Controls("Field" & NameCol - 2)
45                 NameFld.Value = NamesDB.Cells(NameRow, NameCol).Value
46             Next NameCol
47             AddEditName.NameType.Value = NamesDB.Range("B" & NameRow).Value 'Set Name
Type
48             AddEditName.Status.Value = NamesDB.Range("J" & NameRow).Value 'Status
49         End If
50         AddEditName.Show 'Display Form
51     End With
52 End Sub

```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, PgQty As Long, LastPageRow As Long,
PageNum As Long
3 Dim StartRow As Long, EndRow As Long, StartDBRow As Long, EndDBRow As Long
4 Dim StmtFile As String, OrderFile As String
5 Dim OutApp As Object, OutMail As Object
6 Sub Statement_Create()
7     Orders.Range("Z10:AF35,Z48:AF73,Z86:AF111").ClearContents
8     With OrdersDB
9         LastRow = .Range("A99999").End(xlUp).Row
10        If LastRow < 4 Then Exit Sub
11        .Range("M4:M" & LastRow).Formula = .Range("M2").Formula 'Bring Down Balance
Formulas
12        .Range("A3:M" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"BM2:BP6"), CopyToRange:=.Range("BR2:BX2"), Unique:=True
13        LastResultRow = .Range("BR99999").End(xlUp).Row
14        If LastResultRow < 3 Then Exit Sub
15        If LastResultRow < 29 Then 'Single Page
16            Orders.Range("Z10:AF" & LastResultRow + 7).Value = .Range("BR3:BX" &
LastResultRow).Value
17            Orders.PageSetup.PrintArea = "$Z$3:$AF$40" 'Set Single Page Print area
18        Else 'Multi-Page
19            PgQty = Int((LastResultRow - 2) / 26) + 1
20            LastPageRow = (PgQty * 38) + 2
21            Orders.PageSetup.PrintArea = "$Z$3:$AF$" & LastPageRow 'Set Multi- Page
Print area
22            StartRow = 10 'Set Initial Start Row
23            EndRow = 35 'Set Intial End Row
24            StartDBRow = 3 'Set Start Database Row
25            EndDBRow = 28 'Set End Database Row
26            'Add Data
27            For PageNum = 1 To PgQty
28                Orders.Range("Z" & StartRow & ":AF" & EndRow).Value = .Range("BR" &
StartDBRow & ":BX" & EndDBRow).Value 'Add Statement Page Data
29                Orders.Rows((PageNum * 38) + 3).PageBreak = xlPageBreakManual
30                StartRow = StartRow + 38 'Update Statement Page Start Row
31                StartDBRow = StartDBRow + 26 'Update DB Start Row
32                If LastResultRow > EndDBRow + 26 Then 'Additional Pages (not last page)
33                    EndDBRow = EndDBRow + 26
34                    EndRow = EndRow + 38
35                Else 'Last Result Row (last Page)
36                    EndRow = EndRow + LastResultRow - EndDBRow + 12 'Set To Last Result Row
37                    EndDBRow = LastResultRow 'Update End DB Row
38                End If
39            Next PageNum
40        End If
41    End With
42 End Sub
43
44
45 Sub Statement_CreateAll()
46     Dim Subj As String, Mesg As String
47     Dim VarRow As Long
48     Dim ResultRow As Long
49     Dim CustName As String
50     With OrdersDB
51         LastRow = .Range("A99999").End(xlUp).Row
52         If LastRow < 4 Then Exit Sub
53         .Range("M4:N" & LastRow).Formula = .Range("M2:N2").Formula 'Bring Down Balance
Formulas

```

```

1 2
54 .Range("A3:O" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "B22:CB3" ), CopyToRange:=.Range("CD2" ), Unique:=True
55 LastResultRow = .Range("CD99999" ).End(xlUp).Row 'No Records
56 If LastResultRow < 3 Then
57     MsgBox "No Statements available for customers balances with " & Orders.Range(
        "D29" ).Value
58     Exit Sub
59 End If
60 For ResultRow = 3 To LastResultRow 'Loop through customers to create statements
61     CustName = .Range("CD" & ResultRow).Value 'Customer Name
62     Orders.Range("D15" ).Value = CustName
63     Orders.Calculate
64     Statement_Create 'Create Statement
65     If Application.Caller = "PrintStmnts" Then 'Print Statements
66         Orders.PrintOut , , , False, True, False, , , False
67     Else 'Email Statement
68         StmtFile = ThisWorkbook.Path & "\" & .Range("D15" ).Value &
            "_Statement.pdf"
69         If Dir(StmtFile, vbDirectory) <> "" Then Kill (StmtFile) 'Delete file
            if exists
70         Orders.ExportAsFixedFormat xlTypePDF, StmtFile, , , False
71         Set OutApp = CreateObject("Outlook.Application" )
72         Set OutMail = OutApp.createItem(0)
73         Subj = Orders.Range("AI13" ).Value 'Subject
74         Mesg = Orders.Range("AI14" ).Value 'Email Message
75         For VarRow = 14 To 18
76             Subj = Replace(Subj, Orders.Range("AJ" & VarRow).Value, Format(Orders.
                Range("AK" & VarRow).Value, Orders.Range("AK" & VarRow).NumberFormat))
77             Mesg = Replace(Mesg, Orders.Range("AJ" & VarRow).Value, Format(Orders.
                Range("AK" & VarRow).Value, Orders.Range("AK" & VarRow).NumberFormat))
78         Next VarRow
79         With OutMail
80             .To = Orders.Range("B20" ).Value 'Customer email
81             .attachments.Add StmtFile
82             .Subject = Subj 'Dynamic Subject
83             .body = Mesg 'Dynamic Message
84             .display 'To send use .Send
85         End With
86         On Error GoTo 0
87         Set OutMail = Nothing
88     End If
89 Next ResultRow
90 End With
91 End Sub

92
93 Sub Statement_Email()
94     Dim Subj As String, Mesg As String
95     Dim VarRow As Long
96     With Orders
97         If Orders.Range("B35" ).Value = True Then 'Create PDF for Order
98             .PageSetup.PrintArea = "$E$3:$I$39" 'Set Print Area
99             OrderFile = ThisWorkbook.Path & "\" & .Range("B2" ).Value & "_" & .Range(
                "B5" ).Value & ".pdf"
100            If Dir(OrderFile, vbDirectory) <> "" Then Kill (OrderFile) 'Delete file if
                exists
101            .ExportAsFixedFormat xlTypePDF, OrderFile, , , False
102        End If
103        Statement_Create 'Run Macro to Create Statement
104        StmtFile = ThisWorkbook.Path & "\" & .Range("D15" ).Value & "_Statement.pdf"

```

1 2

```

1 2
105 If Dir(StmntFile, vbDirectory) <> "" Then Kill (StmntFile) 'Delete file if
exists
106 .ExportAsFixedFormat xlTypePDF, StmntFile, , , False
107 Set OutApp = CreateObject("Outlook.Application" )
108 Set OutMail = OutApp.createItem(0)
109 Subj = .Range("AI13" ).Value 'Subject
110 Mesg = .Range("AI14" ).Value 'Email Message
111 For VarRow = 14 To 18
112     Subj = Replace(Subj, .Range("AJ" & VarRow).Value, Format(.Range("AK" &
VarRow).Value, .Range("AK" & VarRow).NumberFormat))
113     Mesg = Replace(Mesg, .Range("AJ" & VarRow).Value, Format(.Range("AK" &
VarRow).Value, .Range("AK" & VarRow).NumberFormat))
114 Next VarRow
115 With OutMail
116     .To = Orders.Range("B20" ).Value 'Customer email
117     .attachments.Add StmntFile
118     If Orders.Range("B35" ).Value = True Then .attachments.Add OrderFile
119     .Subject = Subj 'Dynamic Subject
120     .body = Mesg 'Dynamic Message
121     .display 'To send use .Send
122 End With
123 On Error GoTo 0
124 Set OutMail = Nothing
125 End With
126 End Sub
127
128 Sub Statement_Preview()
129     Statement_Create
130     Orders.PrintPreview True
131 End Sub
132
133
134 Sub Statement_PrintOut()
135     If Orders.Range("B35" ).Value = True Then Order_Print 'Run Macro to print selected
order
136     Statement_Create
137     Orders.PrintOut , , , False, True, False, , , False
138 End Sub

```

```

1 Option Explicit
2 Dim TransRow As Long, TransCol As Long, LastRow As Long, SelRow As Long, LastResultRow
  As Long, ResultRow As Long
3
4 Sub Trans_AddEditName()
5     Dim NameFld As Control
6     Dim NameRow As Long, NameCol As Long
7     Unload AddEditName 'Clear Fields In Form
8     With Transactions
9         If .Range("B8").Value = Empty Then 'New Name
10            AddEditName.Field1.Value = .Range("J6").Value
11            AddEditName.NameType.Value = [VendorNm] 'Set Default to Vendor
12            AddEditName.Status.Value = "Active" 'Set Default to Active Status
13        Else 'Existing Name
14            NameRow = .Range("B8").Value 'Set NameRow
15            For NameCol = 3 To 9
16                Set NameFld = AddEditName.Controls("Field" & NameCol - 2)
17                NameFld.Value = NamesDB.Cells(NameRow, NameCol).Value
18            Next NameCol
19            AddEditName.NameType.Value = NamesDB.Range("B" & NameRow).Value 'Set Name
20            Type
21            AddEditName.Status.Value = NamesDB.Range("J" & NameRow).Value 'Status
22        End If
23        AddEditName.Show 'Display Form
24    End With
25 End Sub
26
27
28
29
30 Sub Trans_ClearFilter()
31     Transactions.Range("F5").Value = "Enter Search"
32 End Sub
33
34 Sub Trans_Delete()
35     If MsgBox("Are you sure you want to delete this transaction?" , vbYesNo, "Delete
  Transaction" ) = vbNo Then Exit Sub
36     With Transactions
37         If .Range("B4").Value = Empty Then GoTo NotSaved
38         TransRow = .Range("B4").Value 'Trans Row
39         TransDB.Range(TransRow & ":" & TransRow).EntireRow.Delete 'Delete Transaction
40         NotSaved:
41         Trans_New 'Clear Out All Form Fields
42         Trans_LoadList 'Reload Trans list
43     End With
44 End Sub
45
46 Sub Trans_Load()
47     With Transactions
48         .Range("J4,J6:M6,J8,M8,J10:M10").ClearContents
49         On Error Resume Next
50         If .Range("B4").Value = Empty Then
51             MsgBox "Please make sure to select a correct Transaction"
52             Exit Sub
53         End If
54         Application.ScreenUpdating = False
55         TransRow = .Range("B4").Value 'Trans Row
56         'Load Trans details
57         For TransCol = 2 To 7
58             .Range(TransDB.Cells(1, TransCol).Value).Value = TransDB.Cells(TransRow,

```

1 2 3


```

1 2 3
    TransCol).Value 'Load Transaction Details
59     Next TransCol
60     End With
61     Application.ScreenUpdating = True
62 End Sub
63
64 Sub Trans_LoadList()
65     Transactions.Range("B6,C7:G9999" ).ClearContents 'Clear Existing Transactions
66     With TransDB
67         LastRow = .Range("A9999" ).End(xlUp).Row
68         If LastRow < 4 Then Exit Sub
69         .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
70             "L2:L3" ), CopyToRange:=.Range("O2:S2" ), Unique:=True
71         LastResultRow = .Range("O9999" ).End(xlUp).Row
72         If LastResultRow < 3 Then Exit Sub
73         Transactions.Range("C7:G" & LastResultRow + 4).Value = .Range("O3:S" &
74             LastResultRow).Value 'Bring Transactions over
75         If Transactions.Range("B4" ).Value = Empty Then Exit Sub
76         On Error Resume Next
77         Transactions.Range("B6" ).Value = Transactions.Range("C7:C9999" ).Find(
78             Transactions.Range("B3" ).Value, , xlFormulas, xlWhole).Row
79         On Error GoTo 0
80         If Transactions.Range("B6" ).Value <> Empty Then Transactions.Range("D" &
81             Transactions.Range("B6" ).Value).Select 'Select Transaction item to load
82         transaction
83     End With
84 End Sub
85
86 Sub Trans_New()
87     With Transactions
88         .Range("B3,B6,J4,J6:M6,J8,M8,J10:M10" ).ClearContents
89         .Range("J4" ).Value = Date 'Set current date as default
90         .Range("J8" ).Value = [DefTransAcct] 'Set Default trans. acctout
91         .Range("J6" ).Select
92     End With
93 End Sub
94
95 Sub Trans_SavedMsg()
96     With Transactions.Shapes("TransSavedMsg" )
97         Dim i As Long, Delay As Double, StartTime As Double
98         .Visible = msoCTrue
99         For i = 1 To 100
100            .Fill.Transparency = i / 100
101            Delay = 0.009
102            StartTime = Timer
103            Do
104                DoEvents
105            Loop While Timer - StartTime < Delay
106        Next i
107        .Visible = msoFalse
108    End With
109 End Sub
110
111 Sub Trans_SaveUpdate()
112     With Transactions
113         If .Range("B2" ).Value <> 5 Then
114             MsgBox "Please make sure to add a Transaction Date, Type, Name, Account &
115                 Amount before saving this transaction"
116             Exit Sub
117         End If
118         If .Range("B8" ).Value = Empty Then

```

```
1 2 3
113 MsgBox "Please add and save " & .Range("J6").Value & " before saving this
      transaction"
114 Trans_AddEditName 'Run Macro to add name
115 If .Range("B8").Value = Empty Then Exit Sub
116 End If
117
118 If .Range("B4").Value = Empty Then 'New Trans
119     TransRow = TransDB.Range("A99999").End(xlUp).Row + 1 'New Database Row
120     .Range("B3").Value = .Range("B5").Value 'Set Next Trans ID
121     TransDB.Range("A" & TransRow).Value = .Range("B3").Value 'Set Trans ID
122 Else 'Existing Trans
123     TransRow = .Range("B4").Value 'Existing Trans Row
124 End If
125 'Add/Update Data on App Database
126 For TransCol = 2 To 7
127     TransDB.Cells(TransRow, TransCol).Value = .Range(TransDB.Cells(1, TransCol).
      Value).Value 'Add/Update transaction
128 Next TransCol
129 Trans_LoadList 'Refun Transaction Load
130 Trans_SavedMsg 'Run Fade Out Message
131 End With
132 End Sub
```

—, 11, 12

A

Activate, 18, 29
 ActiveSheet, 17, 29
 Add, 18, 20, 21, 23, 24, 26-30, 35, 36
 AddEditName, 17, 33, 37
 Admin, 5, 11, 12, 18, 26, 27, 29, 30
 AdvancedFilter, 19, 20, 24, 27, 29, 34, 35, 38
 AllowMultiSelect, 18, 21
 Application, 18, 21, 22, 24-26, 28, 29, 35, 37, 38
 Apply, 18, 20, 23, 24, 27
 attachments, 28, 35, 36

B

body, 28, 35, 36
 BrowseForPicFolder, 18, 21, 23
 BuildListOfAccounts, 3, 18

C

Calculate, 5, 11, 35
 Caller, 18, 26, 29, 35
 CancelBtn_Click, 17
 Cell, 3, 4
 Cells, 3, 4, 17, 22, 23, 27, 28, 30, 31, 33, 37, 39
 Chr, 3, 4, 26, 27, 30
 Clear, 18, 20, 23, 24, 27
 ClearContents, 3, 4, 11, 12, 18-24, 26, 28, 29, 34, 37, 38
 CodeName, 17
 Column, 3, 4, 27, 30
 Control, 17, 33, 37
 Controls, 17, 33, 37
 CopyToRange, 19, 20, 24, 27, 29, 34, 35, 38
 CountLarge, 3, 7, 15
 createItem, 28, 35, 36
 CreateObject, 28, 35, 36
 CriteriaRange, 19, 20, 24, 27, 29, 34, 35, 38
 CustName, 34, 35
 CustomerNm, 33

D

Dashboard, 19, 20
 Dashboard_Refresh, 5, 19
 DataOption, 18, 23, 24
 DeffFootMsgRow, 26, 27, 30
 DefStatusRow, 26, 27, 30
 DefTransAcct, 38
 Delay, 22, 31, 38
 Delete, 12, 19, 21-23, 26-29, 37
 Dir, 20, 21, 23, 27, 35, 36
 display, 28, 35, 36
 DoEvents, 22, 31, 38
 DtRange, 5, 11

E

Empty, 3-5, 11, 12, 15, 17, 21, 22, 28, 31, 33, 37-39
 EndDBRow, 34
 EndRow, 34
 EntireRow, 12, 21, 27, 37
 EstimOrder, 11, 26, 28, 30
 Explicit, 3, 5-11, 13-19, 21, 24, 26, 33, 34, 37
 ExportAsFixedFormat, 28, 35, 36

F

Field1, 17, 33, 37
 FileCopy, 21
 FileDialog, 18, 21
 FileName, 26-28
 Fill, 22, 31, 38
 Filters, 21
 Find, 5, 11, 12, 26, 27, 29, 30, 38
 Format, 28, 35, 36
 Formula, 19, 20, 34
 Formula1, 26, 28-30
 FoundItem, 11, 33

H

Height, 20, 23
 Hide, 17
 Home, 29

I

i, 22, 31, 38
 IDRow, 26
 Insert, 20, 23
 InStr, 19
 Int, 34
 Intersect, 3, 5, 7, 11, 12, 15
 InvIncAcct, 26, 30
 InvOrder, 11, 26, 28-30
 Item_BrowseItemPic, 21
 Item_ClearItemPic, 21
 Item_Delete, 21
 Item_Load, 7, 21
 Item_New, 21, 22
 Item_SavedMsg, 22, 23
 Item_SaveUpdate, 22
 Item_ShowItemPic, 21-23
 ItemCol, 21-23
 ItemDbRow, 12
 ItemPic, 21
 ItemPicFolder, 20, 21, 23
 ItemRow, 11, 12, 21-23, 26, 29, 31-33
 Items, 21-23
 Items_UpdateSortedNames, 21, 23
 ItemsDB, 11, 12, 19-23

K

Key, 18, 20, 23, 24, 27
 Kill, 21, 27, 35, 36

L

LastExpRow, 18
 LastIncRow, 18
 LastItemRow, 26, 31, 33
 LastOrderItemDBRow, 26, 27, 29, 33
 LastPageRow, 34
 LastResultRow, 19-21, 24, 26, 27, 29, 33-35, 37, 38
 LastRow, 3, 18-21, 23, 24, 34, 35, 37, 38
 LastTransResultRow, 19
 Left, 20, 23
 LeftPos, 19, 20
 Len, 20
 LockAspectRatio, 23

M

Menu_GoToSheet, 18
 Msg, 27, 28, 34-36
 MsgBox, 11, 12, 17, 21-23, 27, 28, 31, 35, 37-39
 msoCTrue, 22, 23, 31, 33, 38

msoFalse, 12, 22, 26, 29, 31, 33, 38
 msoFileDialogFilePicker, 21
 msoFileDialogFolderPicker, 18

N

Name, 19, 20, 23, 26, 29
 NameCol, 17, 33, 37
 NameFld, 17, 33, 37
 NameRow, 11, 12, 17, 33, 37
 Names_UpdateAndSort, 17, 24
 NamesDB, 11, 12, 17, 24, 33, 37
 NameType, 17, 33, 37
 NoCust, 24
 NoDBItems, 27, 29
 NoEmpl, 24
 Noltems, 19, 20, 31, 32
 NoOrders, 19, 20
 NoSelection, 18, 21
 NoSort, 20, 27
 NoTrans, 19
 NotSaved, 21, 27, 37
 NoVend, 24
 NumberFormat, 28, 35, 36

O

Offset, 3
 OldType, 3
 OrdAcct, 26, 31, 32
 OrdCol, 26, 28, 31, 33
 OrdDate, 26, 31
 OrdDBRow, 26
 Order, 18, 20, 23, 24, 27
 Order_AddNewMenu, 33
 Order_CopyTo, 26
 Order_CopyToMenu, 33
 Order_Delete, 27
 Order_Email, 27
 Order_Load, 12, 28
 Order_New, 27, 29
 Order_Print, 30, 36
 Order_SavedMsg, 30, 32
 Order_SaveUpdate, 31
 OrderFile, 34-36
 OrderItemDBRow, 26, 27, 31-33
 OrderItemRow, 26
 OrderItemsDB, 3, 12, 27, 29, 31, 32
 OrderRow, 3
 Orders, 17, 26-31, 33-36
 Orders_AddEditName, 33
 OrdersDB, 3, 19, 20, 27, 28, 31, 34
 OrderType, 3, 11, 26-31
 OrderTypeNum, 26, 27, 29, 30
 OrdID, 26, 31
 OrdNumb, 26, 33
 OrdRow, 26-28, 31, 33
 OrdShp, 33
 OrdStatus, 26, 31, 32
 OrdType, 33
 OutApp, 26, 28, 34-36
 OutMail, 26, 28, 34-36

P

PageBreak, 34
 PageNumb, 34
 PageSetup, 27, 30, 34, 35
 Path, 27, 35
 PgQty, 34
 PicFolder, 18-21, 23
 PicPath, 19-21, 23
 PicShp, 19

Pictures, 20, 23
 PrintArea, 27, 30, 34, 35
 PrintOut, 30, 35, 36
 PrintPreview, 36
 PurchExpAcct, 26, 30
 PurchOrder, 11, 26, 28, 30, 33

R

Range, 3-5, 7, 11, 12, 15, 17-24, 26-39
 RangeRow, 5, 11
 RefreshAll, 20
 Replace, 28, 29, 35, 36
 ResultRow, 19-21, 26, 27, 29, 33-35, 37
 Row, 3-5, 11, 12, 15, 17-20, 22-24, 26, 27,
 29-31, 34, 35, 38, 39
 Rows, 34

S

SaveBtn_Click, 17
 ScreenUpdating, 18, 22, 24, 25, 28, 29, 37, 38
 SelectedItems, 18, 21
 SelRow, 37
 SetRange, 18, 20, 23, 24, 27
 Shape, 19, 33
 Shapes, 12, 19-23, 26, 29, 31, 33, 38
 Sheets, 18
 Show, 18, 21, 33, 37
 SkipCopy, 21
 Sort, 18, 20, 23, 24, 27
 SortFields, 18, 20, 23, 24, 27
 SortOn, 18, 23, 24
 StartDBRow, 34
 StartRow, 34
 StartTime, 22, 31, 38
 Statement_Create, 34-36
 Statement_CreateAll, 34
 Statement_Email, 35
 Statement_Preview, 36
 Statement_PrintOut, 36
 Status, 17, 33, 37
 StatusType, 26-30
 StmtFile, 34-36
 Subj, 27, 28, 34-36
 Subject, 28, 35, 36

T

Target, 3-5, 7, 11, 12, 15
 TermRow, 12, 26, 30
 Text, 26, 29
 TextFrame2, 26, 29
 TextRange, 26, 29
 ThisWorkbook, 20, 27, 35
 Timer, 22, 31, 38
 Title, 18, 21
 Top, 20, 23
 TopPos, 19, 20
 Trans_AddEditName, 37, 39
 Trans_ClearFilter, 37
 Trans_Delete, 37
 Trans_Load, 15, 37
 Trans_LoadList, 15, 37-39
 Trans_New, 37, 38
 Trans_SavedMsg, 38, 39
 Trans_SaveUpdate, 38
 Transactions, 17, 18, 37, 38
 TransCol, 37-39
 TransDB, 19, 37-39
 Transparency, 22, 31, 38
 TransRow, 37, 39
 TypeNumb, 26

TypeRow, 26, 29

U

Unique, 19, 20, 24, 27, 29, 34, 35, 38
 Unload, 33, 37

V

Validation, 26, 28-30
 Value, 3-5, 7, 11, 12, 15, 17-24, 26-39
 VarRow, 27, 28, 34-36
 vbDirectory, 20, 21, 23, 27, 35, 36
 vbNo, 21, 27, 37
 vbYesNo, 21, 27, 37
 VendorNm, 33, 37
 Visible, 12, 22, 23, 26, 29, 31, 33, 38

W

Width, 20, 23
 Worksheet_Change, 3, 5, 7, 11, 15
 Worksheet_SelectionChange, 3, 7, 12, 15
 WrkOrder, 11, 26, 28, 30

X

xlAscending, 18, 20, 23, 24
 xlDescending, 20, 27
 xlFilterCopy, 19, 20, 24, 27, 29, 34, 35, 38
 xlFormulas, 38
 xlPageBreakManual, 34
 xlSortNormal, 18, 23, 24
 xlSortOnValues, 18, 23, 24
 xlTypePDF, 28, 35, 36
 xlUp, 3, 17-20, 22-24, 27, 29, 31, 34, 35, 38,
 39
 xlValidAlertInformation, 26, 28-30
 xlValidateList, 26, 28-30
 xlValues, 5, 11, 12, 26, 27, 29, 30
 xlWhole, 5, 11, 12, 26, 27, 29, 30, 38

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