

VBA SOURCE CODE BOOK

MULTI-LINGUAL INTEGRATION

FREE DOWNLOAD



Excel For Freelancers

Excel Mastery: Make Your Application Multilingual In Minutes + FREE DOWNLOAD



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](#)



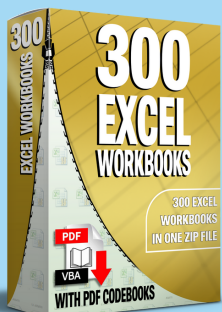
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[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.

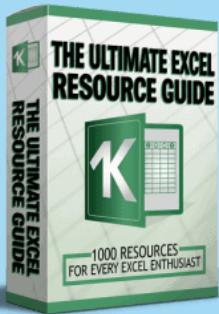
[Click here to learn more](#)



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](#)

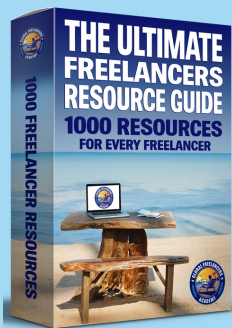
OUR COURSES & PRODUCTS



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.....	6
VBAProject.....	6
Documents.....	6
Admin.....	6
(Declarations).....	6
Worksheet_Change [Sub].....	6
ChartData.....	7
(Declarations).....	7
CustomerDB.....	8
(Declarations).....	8
Worksheet_Change [Sub].....	8
Customers.....	9
(Declarations).....	9
Worksheet_Change [Sub].....	9
Worksheet_SelectionChange [Sub].....	9
Dashboard.....	10
(Declarations).....	10
EmployeeDB.....	11
(Declarations).....	11
Worksheet_Change [Sub].....	11
Employees.....	12
(Declarations).....	12
Worksheet_Change [Sub].....	12
Worksheet_SelectionChange [Sub].....	12
LangDB.....	13
(Declarations).....	13
OrderDB.....	14
(Declarations).....	14
Worksheet_Change [Sub].....	14
OrderItems.....	15
(Declarations).....	15
Worksheet_Change [Sub].....	15
Orders.....	16
(Declarations).....	16
Worksheet_Change [Sub].....	16
Worksheet_SelectionChange [Sub].....	17
PaidEmpl.....	18
(Declarations).....	18
PayItems.....	19
(Declarations).....	19
Payroll.....	20
(Declarations).....	20
Worksheet_Change [Sub].....	20
Worksheet_SelectionChange [Sub].....	20
PayrollDB.....	21
(Declarations).....	21
Worksheet_Change [Sub].....	21
ProductDB.....	22
(Declarations).....	22
Worksheet_Change [Sub].....	22
Products.....	23
(Declarations).....	23
Worksheet_Change [Sub].....	23
Worksheet_SelectionChange [Sub].....	23
ThisWorkbook.....	24
(Declarations).....	24
Timesheet.....	25
(Declarations).....	25
Worksheet_Change [Sub].....	25
Transactions.....	26
(Declarations).....	26
Worksheet_Change [Sub].....	26
Worksheet_SelectionChange [Sub].....	26
TransDB.....	27
(Declarations).....	27
Worksheet_Change [Sub].....	27

UserDB	28
(Declarations)	28
Worksheet_Change [Sub]	28
Users	29
(Declarations)	29
Worksheet_Change [Sub]	29
Worksheet_SelectionChange [Sub]	29
VendorDB	30
(Declarations)	30
Worksheet_Change [Sub]	30
Vendors	31
(Declarations)	31
Worksheet_Change [Sub]	31
Worksheet_SelectionChange [Sub]	31
Forms	32
CustomerFrm	32
(Declarations)	32
CancelBtn_Click [Sub]	32
SaveBtn_Click [Sub]	32
EmployeeFrm	33
(Declarations)	33
BrowseBtn_Click [Sub]	33
CancelBtn_Click [Sub]	33
Field3_Change [Sub]	33
SaveBtn_Click [Sub]	33
LoginFrm	34
(Declarations)	34
CancelBtn_Click [Sub]	34
OkBtn_Click [Sub]	34
OrdersFrm	35
(Declarations)	35
EnableCheckboxes_Click [Sub]	35
InclInvoice_Click [Sub]	35
IncPurch_Click [Sub]	35
IncWO_Click [Sub]	35
PrintBtn_Click [Sub]	35
SettingImageList [Sub]	35
TreeView1_Click [Sub]	35
UpdateOrdersBtn_Click [Sub]	36
UserForm_Activate [Sub]	36
OrdersTreeView	37
(Declarations)	37
Checkboxes_Click [Sub]	37
CommandButton1_Click [Sub]	37
InclInvoices_Click [Sub]	37
IncPurchases_Click [Sub]	37
IncWorkOrders_Click [Sub]	37
PrintBtn_Click [Sub]	37
SettingImages [Sub]	37
TreeView2_Click [Sub]	37
UserForm_Activate [Sub]	37
ProductFrm	39
(Declarations)	39
BrowseBtn_Click [Sub]	39
CancelBtn_Click [Sub]	39
SaveBtn_Click [Sub]	39
ResourceFrm	40
(Declarations)	40
BrowseBtn_Click [Sub]	40
CancelBtn_Click [Sub]	40
ResType_Change [Sub]	40
SaveBtn_Click [Sub]	40
TransactFrm	41
(Declarations)	41
CancelBtn_Click [Sub]	41
Field3_Change [Sub]	41

SaveBtn_Click [Sub]	41
UserFrm	42
(Declarations)	42
BrowseBtn_Click [Sub]	42
CancelBtn_Click [Sub]	42
SaveBtn_Click [Sub]	42
VendorFrm	43
(Declarations)	43
CancelBtn_Click [Sub]	43
CancelBtn_Exit [Sub]	43
SaveBtn_Click [Sub]	43
UserForm_QueryClose [Sub]	43
Modules	44
Admin_Macros	44
(Declarations)	44
BrowseForEmployeePicFolder [Sub]	44
BrowseForMainSharedFolder [Sub]	44
BrowseForOrderPicFolder [Sub]	44
BrowseForProductPicFolder [Sub]	44
BrowseForUserPicFolder [Sub]	44
CoAuthoring_Macros	46
(Declarations)	46
SharedFolder_Check [Sub]	46
Workbook_CheckAndSync [Sub]	46
Customer_Macros	48
(Declarations)	48
Customer_AddNew [Sub]	48
Customer_ClearFilter [Sub]	48
Customer_Delete [Sub]	48
Customer_Edit [Sub]	48
Customer_RefreshList [Sub]	48
Customer_SaveUpdate [Sub]	49
Employee_Macros	50
(Declarations)	50
Employee_AddNew [Sub]	50
Employee_AddPicture [Sub]	50
Employee_AddUpdatePicture [Sub]	50
Employee_ClearFilter [Sub]	50
Employee_Delete [Sub]	51
Employee_Edit [Sub]	51
Employee_RefreshList [Sub]	51
EmployeeList_SaveUpdate [Sub]	52
EmployeePayroll_SaveUpdate [Sub]	52
Language_Macros	54
(Declarations)	54
ChangeLanguage [Sub]	54
GetButtonText [Sub]	56
GetLabelValues [Sub]	56
GetUserformLabelsAndButtons [Sub]	57
Reset [Sub]	57
ShowVendForm [Sub]	57
Translate [Function]	57
LoginOut_Macros	59
(Declarations)	59
UserLogin [Sub]	59
UserLogin_Show [Sub]	59
UserLogout [Sub]	59
Order_Macros	61
(Declarations)	61
CreateAllOrderImages [Sub]	61
CreateOrderPicture [Sub]	61
Order_AddNew [Sub]	61
Order_Delete [Sub]	62
Order_Load [Sub]	62
Order_Print [Sub]	63
Order_SavedMsg [Sub]	63

Order_SaveUpdate [Sub].....	63
Orders_List [Sub].....	64
OrderTab_Invoice [Sub].....	65
OrderTab_PO [Sub].....	65
OrderTab_WO [Sub].....	66
PayEmployee_Macros	67
(Declarations)	67
PayEmpl_Load [Sub]	67
PayEmpl_LoadPayItems [Sub].....	67
PayEmpl_LoadTimesheetEntries [Sub].....	69
PayEmpl_Previous [Sub]	69
PayEmpl_SaveNext [Sub].....	69
Payroll_Macros	73
(Declarations)	73
Payroll_AddAllEmployees [Sub].....	73
Payroll_AddNew [Sub]	73
Payroll_LoadList [Sub]	73
Payroll_LoadPayroll [Sub].....	74
Payroll_SaveUpdate [Sub].....	75
PrintPayStub [Sub]	75
Product_Macros	76
(Declarations)	76
Product_AddNew [Sub].....	76
Product_AddPicture [Sub].....	76
Product_ClearFilter [Sub].....	76
Product_Delete [Sub].....	76
Product_Edit [Sub].....	76
Product_RefreshList [Sub]	77
Product_SaveUpdate [Sub]	77
Resource_Macros	79
(Declarations)	79
Resource_AddNew [Sub].....	79
Resource_AddPicture [Sub].....	79
Resource_Edit [Sub].....	79
Resource_SaveUpdate [Sub].....	80
Toolbar_Macros	81
(Declarations)	81
Toolbar_Admin [Sub]	81
Toolbar_Customers [Sub]	81
Toolbar_Dashboard [Sub].....	81
Toolbar_Employees [Sub].....	81
Toolbar_Invoice [Sub].....	81
Toolbar_Payroll [Sub]	81
Toolbar_Products [Sub]	81
Toolbar_PurchaseOrders [Sub]	81
Toolbar_Transactions [Sub].....	81
Toolbar_Users [Sub].....	81
Toolbar_Vendors [Sub].....	81
Toolbar_WorkOrders [Sub].....	81
TrainTreeView.....	82
(Declarations)	82
GenerateTreeView [Sub]	82
PrintSelectedOrders [Sub]	83
ShowTrainTreeView [Sub]	83
Transaction_Macros.....	84
(Declarations)	84
Transaction_AddNew [Sub]	84
Transaction_ClearFilter [Sub]	84
Transaction_Delete [Sub]	84
Transaction_Edit [Sub]	84
Transaction_RefreshList [Sub].....	84
Transaction_SaveUpdate [Sub]	85
TreeView_Macros	86
(Declarations)	86
CreateTreeView [Sub].....	86
PrintSelectedOrdersOld [Sub].....	87

Table of Contents

ShowTreeView [Sub]	87
User_Macros.....	88
(Declarations)	88
User_AddNew [Sub]	88
User_AddPicture [Sub]	88
User_ClearFilter [Sub]	88
User_Delete [Sub]	88
User_Edit [Sub].....	88
User_RefreshList [Sub].....	89
User_SaveUpdate [Sub]	89
Vendor_Macros.....	90
(Declarations)	90
Vendor_AddNew [Sub]	90
Vendor_ClearFilter [Sub]	90
Vendor_Delete [Sub].....	90
Vendor_Edit [Sub].....	90
Vendor_RefreshList [Sub].....	90
Vendor_SaveUpdate [Sub]	91

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("R7" )) Is Nothing Then ChangeLanguage
5 End Sub
6
7
```

1 Option Explicit
2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:I999999")) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12     Application.EnableEvents = False
13     Application.Undo
14     Application.EnableEvents = True
15     End
16 End If
17 If [ShareSyncRun] = False Then 'Run When Sync is not active
18     SharedFolder_Check 'Check For Shared Folder
19     SharedFold = [SharedFolder] 'Set Shared Folder
20     CurrentUser = [LoggedInUser] 'Current User Name
21     SheetNm = Target.Worksheet.Name 'Sheet Name
22     CellAdd = Target.Address 'Cell Address
23     CellVal = Target.Value 'Cell Value
24     LastUserRow = UserDB.Range("A9999").End(xlUp).Row 'Last User Row
25     If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26     For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27     respective folders (except current user)
28     Username = UserDB.Range("B" & UserRow).Value 'User Name
29     If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30     current users folder
31     If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32     SharedFold & "\User Changes" )
33     If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34     Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35     it does not exist
36     FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37     "--" & CellAdd & ".txt" 'Create Text File Name
38     TextFile = FreeFile 'Assign Unique & Free File #
39     Open FilePath For Output As TextFile
40     Print #TextFile, CurrentUser 'Add Current User
41     Print #TextFile, CellVal 'Write Cell Value
42     Close #TextFile 'Close Text file
43     NextUser:
44     Next UserRow
45 End If
46 End If
47 End Sub
```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F2" )) Is Nothing Then Customer_RefreshList
5 End Sub
6
7 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
8     If Target.CountLarge > 1 Then Exit Sub
9     If Shapes("EditGrp" ).Visible = True Then Shapes("EditGrp" ).Visible = msoFalse
10    'Hide Edit Group if visible
11    If Not Intersect(Target, Range("D4:K99999" )) Is Nothing And Range("D" & Target.Row
12    ).Value <> Empty Then
13        Range("B3" ).Value = Target.Row 'Set Selected Row
14        Range("B4" ).Value = Range("L" & Target.Row).Value 'Set DB Row
15        With Shapes("EditGrp" )
16            .Left = Range("L" & Target.Row).Left
17            .Top = Range("L" & Target.Row).Top
18            .Visible = msoCTrue
19        End With
20    End If
21 End Sub
```

1
2

Option Explicit

```
1 Option Explicit
```

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

```
4 Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5 , FileText As String, OldValue As String
```

```
6 Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7 CurrentUser As String
```

```
8 Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
```

```
9 If Not Intersect(Target, Range("A4:O999999" )) Is Nothing And [ShareSyncEnabled] =
```

```
10 "Yes" Then 'Change this range based on your data range or table
```

```
11 If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
```

```
12 Application.EnableEvents = False
```

```
13 Application.Undo
```

```
14 Application.EnableEvents = True
```

```
15 End
```

```
16 End If
```

```
17 If [ShareSyncRun] = False Then 'Run When Sync is not active
```

```
18 SharedFolder_Check 'Check For Shared Folder
```

```
19 SharedFold = [SharedFolder] 'Set Shared Folder
```

```
20 CurrentUser = [LoggedInUser] 'Current User Name
```

```
21 SheetNm = Target.Worksheet.Name 'Sheet Name
```

```
22 CellAdd = Target.Address 'Cell Address
```

```
23 CellVal = Target.Value 'Cell Value
```

```
24 LastUserRow = UserDB.Range("A9999" ).End(xlUp).Row 'Last User Row
```

```
25 If LastUserRow < 3 Then Exit Sub 'Exit On No Users
```

```
26 For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27 respective folders (except current user)
```

```
28 Username = UserDB.Range("B" & UserRow).Value 'User Name
```

```
29 If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30 current users folder
```

```
31 If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32 SharedFold & "\User Changes" )
```

```
33 If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34 Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35 it does not exist
```

```
36 FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37 "--" & CellAdd & ".txt" 'Create Text File Name
```

```
38 TextFile = FreeFile 'Assign Unique & Free File #
```

```
39 Open FilePath For Output As TextFile
```

```
40 Print #TextFile, CurrentUser 'Add Current User
```

```
41 Print #TextFile, CellVal 'Write Cell Value
```

```
42 Close #TextFile 'Close Text file
```

```
43 NextUser:
```

```
44 Next UserRow
```

```
45 End If
```

```
46 End If
```

```
47 End Sub
```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F2" )) Is Nothing Then Employee_RefreshList
5 End Sub
6
7 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
8     If Target.CountLarge > 1 Then Exit Sub
9     If Shapes("EditGrp" ).Visible = True Then Shapes("EditGrp" ).Visible = msoFalse
10    'Hide Edit Group if visible
11    If Not Intersect(Target, Range("D4:Q99999" )) Is Nothing And Range("D" & Target.Row
12    ).Value <> Empty Then
13        Range("B3" ).Value = Target.Row 'Set Selected Row
14        Range("B4" ).Value = Range("R" & Target.Row).Value 'Set DB Row
15        With Shapes("EditGrp" )
16            .Left = Range("R" & Target.Row).Left
17            .Top = Range("R" & Target.Row).Top
18            .Visible = msoCTrue
19        End With
20    End If
21 End Sub
```


1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:H999999" )) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12     Application.EnableEvents = False
13     Application.Undo
14     Application.EnableEvents = True
15     End
16 End If
17 If [ShareSyncRun] = False Then 'Run When Sync is not active
18     SharedFolder_Check 'Check For Shared Folder
19     SharedFold = [SharedFolder] 'Set Shared Folder
20     CurrentUser = [LoggedInUser] 'Current User Name
21     SheetNm = Target.Worksheet.Name 'Sheet Name
22     CellAdd = Target.Address 'Cell Address
23     CellVal = Target.Value 'Cell Value
24     LastUserRow = UserDB.Range("A9999" ).End(xlUp).Row 'Last User Row
25     If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26     For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27     respective folders (except current user)
28     Username = UserDB.Range("B" & UserRow).Value 'User Name
29     If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30     current users folder
31     If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32     SharedFold & "\User Changes" )
33     If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34     Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35     it does not exist
36     FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37     "--" & CellAdd & ".txt" 'Create Text File Name
38     TextFile = FreeFile 'Assign Unique & Free File #
39     Open FilePath For Output As TextFile
40     Print #TextFile, CurrentUser 'Add Current User
41     Print #TextFile, CellVal 'Write Cell Value
42     Close #TextFile 'Close Text file
43     NextUser:
44     Next UserRow
45 End For
46 End If
47 End Sub
48
49
```

```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:K999999" )) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12     Application.EnableEvents = False
13     Application.Undo
14     Application.EnableEvents = True
15     End
16 End If
17 If [ShareSyncRun] = False Then 'Run When Sync is not active
18     SharedFolder_Check 'Check For Shared Folder
19     SharedFold = [SharedFolder] 'Set Shared Folder
20     CurrentUser = [LoggedInUser] 'Current User Name
21     SheetNm = Target.Worksheet.Name 'Sheet Name
22     CellAdd = Target.Address 'Cell Address
23     CellVal = Target.Value 'Cell Value
24     LastUserRow = UserDB.Range("A9999" ).End(xlUp).Row 'Last User Row
25     If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26     For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27     respective folders (except current user)
28     Username = UserDB.Range("B" & UserRow).Value 'User Name
29     If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30     current users folder
31     If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then MkDir (
32     SharedFold & "\User Changes" )
33     If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34     MkDir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35     it does not exist
36     FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37     "--" & CellAdd & ".txt" 'Create Text File Name
38     TextFile = FreeFile 'Assign Unique & Free File #
39     On Error Resume Next
40     Open FilePath For Output As TextFile
41     Print #TextFile, CurrentUser 'Add Current User
42     Print #TextFile, CellVal 'Write Cell Value
43     Close #TextFile 'Close Text file
44     On Error GoTo 0
45     NextUser:
46     Next UserRow
47 End If
48 End If
49 End Sub

```

```

1 Option Explicit
2
3
4 Private Sub Worksheet_Change(ByVal Target As Range)
5     'On Search Order Change
6     If Not Intersect(Target, Range("Q5" )) Is Nothing And Range("Q5" ).Value <> Empty
7         Then
8             Range("B3" ).Value = Range("Q5" ).Value
9             Order_Load
10        End If
11
12    'On Vendor/Customer Change, Load Address
13    If Not Intersect(Target, Range("K6" )) Is Nothing And Range("K6" ).Value <> Empty
14        Then
15            Dim RecRow As Long, DBSht As Worksheet
16            If Range("B9" ).Value = "" Then Exit Sub
17            RecRow = Range("B9" ).Value 'Cust/Vend Record Row
18            If Range("B2" ).Value = "PURCHASE" Then 'Purchase Order (Vendor)
19                Set DBSht = Worksheets("Vendor DB" )
20            Else 'Invoice or Work Order (Customer)
21                Set DBSht = Worksheets("Customer DB" )
22            End If
23            Range("K7" ).Value = DBSht.Range("C" & RecRow).Value 'Address
24            Range("K8" ).Value = DBSht.Range("D" & RecRow).Value & ", " & DBSht.Range("E"
25                & RecRow).Value & " " & DBSht.Range("F" & RecRow).Value
26        End If
27
28    'On Item Change (But not on Order Load)
29    If Not Intersect(Target, Range("J11:J48" )) Is Nothing And Range("B1" ).Value =
30        False Then
31
32        If Range("J" & Target.Row).Value <> Empty Then
33            Dim ProdRow As Long
34            On Error Resume Next
35            ProdRow = ProductDB.Range("Prod_Name" ).Find(Target.Value, , xlValues, xlWhole
36                ).Row
37            On Error GoTo 0
38            If ProdRow <> 0 Then
39                Range("I" & Target.Row).Value = ProductDB.Range("A" & ProdRow).Value
40                'Prod. ID
41                Range("K" & Target.Row).Value = ProductDB.Range("D" & ProdRow).Value
42                'Description
43                Range("L" & Target.Row).Value = 1 'Set default qty
44                'Set Sales Or Purchase Price
45                If Range("B2" ).Value = "INVOICE" Then 'Invoice type
46                    Range("M" & Target.Row).Value = ProductDB.Range("G" & ProdRow).Value
47                    'Sales Price
48                Else 'Purchase Order
49                    Range("M" & Target.Row).Value = ProductDB.Range("F" & ProdRow).Value
50                    'Purchase Price
51                End If
52            End If
53        End If
54
55    Else 'On Delete, Clear Item Details
56        Range("I" & Target.Row).ClearContents 'Clear Item ID
57        Range("K" & Target.Row & ":M" & Target.Row).ClearContents 'Clear Item
58        Details
59    End If
60 End If
61 End Sub

```

```
52  
53 Private Sub Worksheet_SelectionChange(ByVal Target As Range)  
54     If Target.CountLarge > 1 Then Exit Sub  
55     If Not Intersect(Target, Range("D6:H999" )) Is Nothing And Range("D" & Target.Row).  
        Value <> Empty Then  
56         Range("B3" ).Value = Range("D" & Target.Row).Value 'Order ID  
57         Range("B8" ).Value = Target.Row 'Set Selected Row  
58         Order_Load 'Run Macro to load order  
59     End If  
60 End Sub
```

1
2

Option Explicit

1
2

Option Explicit

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     'On Change of Show Year, reload Payroll list
5     If Not Intersect(Target, Range("E4" )) Is Nothing And Range("E4" ).Value <> Empty
6         Then Payroll_LoadList
7
8     'On Change of Employee filters
9     If Not Intersect(Target, Range("I11,K11" )) Is Nothing Then
10        If Range("B3" ).Value <> "" Then
11            MsgBox "Please filter employees for payroll that has not been saved"
12            Exit Sub
13        End If
14        Payroll_AddAllEmployees 'Run Macro to add employees
15    End If
16 End Sub
17
18 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
19     If Target.CountLarge > 1 Then Exit Sub
20
21     'On Payroll Selection, Load Payroll
22     If Not Intersect(Target, Range("D6:E9999" )) Is Nothing And Range("D" & Target.Row)
23         .Value <> Empty Then
24         Range("B2" ).Value = Range("C" & Target.Row).Value 'Set Payroll ID
25         Range("B5" ).Value = Target.Row 'Se Payroll Row
26         Payroll_LoadPayroll 'Run Macro To load Payroll
27     End If
28
29     'On Employee Selection
30     If Not Intersect(Target, Range("H13:K9999" )) Is Nothing And Range("H" & Target.Row)
31         .Value <> Empty Then
32         If Range("L" & Target.Row).Value = Empty Then
33             MsgBox "Please save the payroll before selecting any employee"
34             Exit Sub
35         End If
36         Range("B13" ).Value = Range("L" & Target.Row).Value 'Paid Empl. DB Row
37         Range("B7" ).Value = Range("F" & Target.Row).Value 'Employee ID
38         Range("B10" ).Value = Target.Row 'Set Selected Row
39         PayEmpl_Load 'Run Macro to Load Employee Pay Details
40     End If
41 End Sub
```



```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileText As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:G999999" )) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12     Application.EnableEvents = False
13     Application.Undo
14     Application.EnableEvents = True
15     End
16 End If
17 If [ShareSyncRun] = False Then 'Run When Sync is not active
18     SharedFolder_Check 'Check For Shared Folder
19     SharedFold = [SharedFolder] 'Set Shared Folder
20     CurrentUser = [LoggedInUser] 'Current User Name
21     SheetNm = Target.Worksheet.Name 'Sheet Name
22     CellAdd = Target.Address 'Cell Address
23     CellVal = Target.Value 'Cell Value
24     LastUserRow = UserDB.Range("A9999" ).End(xlUp).Row 'Last User Row
25     If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26     For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27     respective folders (except current user)
28     Username = UserDB.Range("B" & UserRow).Value 'User Name
29     If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30     current users folder
31     If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32     SharedFold & "\User Changes" )
33     If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34     Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35     it does not exist
36     FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37     "--" & CellAdd & ".txt" 'Create Text File Name
38     TextFile = FreeFile 'Assign Unique & Free File #
39     Open FilePath For Output As TextFile
40     Print #TextFile, CurrentUser 'Add Current User
41     Print #TextFile, CellVal 'Write Cell Value
42     Close #TextFile 'Close Text file
43     NextUser:
44     Next UserRow
45 End If
46 End If
47 End Sub

```

```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:J999999" )) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12     Application.EnableEvents = False
13     Application.Undo
14     Application.EnableEvents = True
15     End
16     End If
17     If [ShareSyncRun] = False Then 'Run When Sync is not active
18     SharedFolder_Check 'Check For Shared Folder
19     SharedFold = [SharedFolder] 'Set Shared Folder
20     CurrentUser = [LoggedInUser] 'Current User Name
21     SheetNm = Target.Worksheet.Name 'Sheet Name
22     CellAdd = Target.Address 'Cell Address
23     CellVal = Target.Value 'Cell Value
24     LastUserRow = UserDB.Range("A9999" ).End(xlUp).Row 'Last User Row
25     If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26     For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27     respective folders (except current user)
28     Username = UserDB.Range("B" & UserRow).Value 'User Name
29     If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30     current users folder
31     If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32     SharedFold & "\User Changes" )
33     If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34     Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35     it does not exist
36     FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37     "--" & CellAdd & ".txt" 'Create Text File Name
38     TextFile = FreeFile 'Assign Unique & Free File #
39     Open FilePath For Output As TextFile
40     Print #TextFile, CurrentUser 'Add Current User
41     Print #TextFile, CellVal 'Write Cell Value
42     Close #TextFile 'Close Text file
43     NextUser:
44     Next UserRow
45     End If
46     End If
47 End Sub

```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     On Error Resume Next
5     If Not Intersect(Target, Range("F2" )) Is Nothing Then Product_RefreshList
6     On Error GoTo 0
7 End Sub
8
9 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
10    If Target.CountLarge > 1 Then Exit Sub
11    If Shapes("EditGrp" ).Visible = True Then Shapes("EditGrp" ).Visible = msoFalse
12    'Hide Edit Group if visible
13    If Not Intersect(Target, Range("D4:L9999" )) Is Nothing And Range("D" & Target.Row
14    ).Value <> Empty Then
15        Range("B3" ).Value = Target.Row 'Set Selected Row
16        Range("B4" ).Value = Range("M" & Target.Row).Value 'Set DB Row
17        With Shapes("EditGrp" )
18            .Left = Range("M" & Target.Row).Left
19            .Top = Range("M" & Target.Row).Top
20            .Visible = msoCTrue
21        End With
22    End If
23 End Sub
```

1 Option Explicit

2

```
1 Option Explicit
```

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

```
4 Dim SharedFold As String, Username As String, FileName As String, FilePath As String  
5 , FileType As String, OldValue As String
```

```
5 Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,  
6 CurrentUser As String
```

```
6 Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
```

```
7 If Not Intersect(Target, Range("A4:G999999" )) Is Nothing And [ShareSyncEnabled] =
```

```
"Yes" Then 'Change this range based on your data range or table
```

```
8 If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
```

```
9 Application.EnableEvents = False
```

```
10 Application.Undo
```

```
11 Application.EnableEvents = True
```

```
12 End
```

```
13 End If
```

```
14 If [ShareSyncRun] = False Then 'Run When Sync is not active
```

```
15 SharedFolder_Check 'Check For Shared Folder
```

```
16 SharedFold = [SharedFolder] 'Set Shared Folder
```

```
17 CurrentUser = [LoggedInUser] 'Current User Name
```

```
18 SheetNm = Target.Worksheet.Name 'Sheet Name
```

```
19 CellAdd = Target.Address 'Cell Address
```

```
20 CellVal = Target.Value 'Cell Value
```

```
21 LastUserRow = UserDB.Range("A9999" ).End(xlUp).Row 'Last User Row
```

```
22 If LastUserRow < 3 Then Exit Sub 'Exit On No Users
```

```
23 For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their  
24 respective folders (except current user)
```

```
24 Username = UserDB.Range("B" & UserRow).Value 'User Name
```

```
25 If CurrentUser = Username Then GoTo NextUser 'No need to add changes into  
26 current users folder
```

```
26 If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (  
27 SharedFold & "\User Changes" )
```

```
27 If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then  
28 Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if  
29 it does not exist
```

```
28 FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &  
29 "--" & CellAdd & ".txt" 'Create Text File Name
```

```
29 TextFile = FreeFile 'Assign Unique & Free File #
```

```
30 Open FilePath For Output As TextFile
```

```
31 Print #TextFile, CurrentUser 'Add Current User
```

```
32 Print #TextFile, CellVal 'Write Cell Value
```

```
33 Close #TextFile 'Close Text file
```

```
34 NextUser:
```

```
35 Next UserRow
```

```
36 End If
```

```
37 End If
```

```
38 End Sub
```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F2" )) Is Nothing Then Transaction_RefreshList
5 End Sub
6
7 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
8     If Target.CountLarge > 1 Then Exit Sub
9     If Shapes("EditGrp" ).Visible = True Then Shapes("EditGrp" ).Visible = msoFalse
10    'Hide Edit Group if visible
11    If Not Intersect(Target, Range("D4:J99999" )) Is Nothing And Range("D" & Target.Row
12    ).Value <> Empty Then
13        Range("B3" ).Value = Target.Row 'Set Selected Row
14        Range("B4" ).Value = Range("K" & Target.Row).Value 'Set DB Row
15        With Shapes("EditGrp" )
16            .Left = Range("K" & Target.Row).Left
17            .Top = Range("K" & Target.Row).Top
18            .Visible = msoCTrue
19        End With
20    End If
21 End Sub
```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:H999999" )) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12     Application.EnableEvents = False
13     Application.Undo
14     Application.EnableEvents = True
15     End
16 End If
17 If [ShareSyncRun] = False Then 'Run When Sync is not active
18     SharedFolder_Check 'Check For Shared Folder
19     SharedFold = [SharedFolder] 'Set Shared Folder
20     CurrentUser = [LoggedInUser] 'Current User Name
21     SheetNm = Target.Worksheet.Name 'Sheet Name
22     CellAdd = Target.Address 'Cell Address
23     CellVal = Target.Value 'Cell Value
24     LastUserRow = UserDB.Range("A9999" ).End(xlUp).Row 'Last User Row
25     If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26     For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27     respective folders (except current user)
28     Username = UserDB.Range("B" & UserRow).Value 'User Name
29     If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30     current users folder
31     If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32     SharedFold & "\User Changes" )
33     If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34     Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35     it does not exist
36     FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37     "--" & CellAdd & ".txt" 'Create Text File Name
38     TextFile = FreeFile 'Assign Unique & Free File #
39     Open FilePath For Output As TextFile
40     Print #TextFile, CurrentUser 'Add Current User
41     Print #TextFile, CellVal 'Write Cell Value
42     Close #TextFile 'Close Text file
43     NextUser:
44     Next UserRow
45 End If
46 End If
47 End Sub
```

```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:F999999")) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12     Application.EnableEvents = False
13     Application.Undo
14     Application.EnableEvents = True
15     End
16     End If
17     If [ShareSyncRun] = False Then 'Run When Sync is not active
18     SharedFolder_Check 'Check For Shared Folder
19     SharedFold = [SharedFolder] 'Set Shared Folder
20     CurrentUser = [LoggedInUser] 'Current User Name
21     SheetNm = Target.Worksheet.Name 'Sheet Name
22     CellAdd = Target.Address 'Cell Address
23     CellVal = Target.Value 'Cell Value
24     LastUserRow = UserDB.Range("A9999").End(xlUp).Row 'Last User Row
25     If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26     For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27     respective folders (except current user)
28     Username = UserDB.Range("B" & UserRow).Value 'User Name
29     If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
30     current users folder
31     If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32     SharedFold & "\User Changes" )
33     If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
34     Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
35     it does not exist
36     FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
37     "--" & CellAdd & ".txt" 'Create Text File Name
38     TextFile = FreeFile 'Assign Unique & Free File #
39     Open FilePath For Output As TextFile
40     Print #TextFile, CurrentUser 'Add Current User
41     Print #TextFile, CellVal 'Write Cell Value
42     Close #TextFile 'Close Text file
43     NextUser:
44     Next UserRow
45     End If
46     End If
47 End Sub

```



```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F2" )) Is Nothing Then User_RefreshList
5 End Sub
6
7 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
8     If Target.CountLarge > 1 Then Exit Sub
9     If Shapes("EditGrp" ).Visible = True Then Shapes("EditGrp" ).Visible = msoFalse
10    'Hide Edit Group if visible
11    If Not Intersect(Target, Range("D4:H99999" )) Is Nothing And Range("D" & Target.Row
12    ).Value <> Empty Then
13        Range("B3" ).Value = Target.Row 'Set Selected Row
14        Range("B4" ).Value = Range("H" & Target.Row).Value 'Set DB Row
15        With Shapes("EditGrp" )
16            .Left = Range("H" & Target.Row).Left
17            .Top = Range("H" & Target.Row).Top
18            .Visible = msoCTrue
19        End With
20    End If
21 End Sub
```

```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, Username As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:I999999")) Is Nothing And [ShareSyncEnabled] =
10     "Yes" Then 'Change this range based on your data range or table
11     If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12         Application.EnableEvents = False
13         Application.Undo
14         Application.EnableEvents = True
15     End If
16     If [ShareSyncRun] = False Then 'Run When Sync is not active
17         SharedFolder_Check 'Check For Shared Folder
18         SharedFold = [SharedFolder] 'Set Shared Folder
19         CurrentUser = [LoggedInUser] 'Current User Name
20         SheetNm = Target.Worksheet.Name 'Sheet Name
21         CellAdd = Target.Address 'Cell Address
22         CellVal = Target.Value 'Cell Value
23         LastUserRow = UserDB.Range("A9999").End(xlUp).Row 'Last User Row
24         If LastUserRow < 3 Then Exit Sub 'Exit On No Users
25         For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
26         respective folders (except current user)
27             Username = UserDB.Range("B" & UserRow).Value 'User Name
28             If CurrentUser = Username Then GoTo NextUser 'No need to add changes into
29             current users folder
30             If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
31             SharedFold & "\User Changes" )
32             If Dir(SharedFold & "\User Changes\" & Username, vbDirectory) = "" Then
33             Mkdir (SharedFold & "\User Changes\" & Username) 'Create user Folder if
34             it does not exist
35             FilePath = SharedFold & "\User Changes\" & Username & "\" & SheetNm &
36             "--" & CellAdd & ".txt" 'Create Text File Name
37             TextFile = FreeFile 'Assign Unique & Free File #
38             Open FilePath For Output As TextFile
39             Print #TextFile, CurrentUser 'Add Current User
40             Print #TextFile, CellVal 'Write Cell Value
41             Close #TextFile 'Close Text file
42             NextUser:
43         Next UserRow
44     End If
45 End If
46 End Sub

```

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F2" )) Is Nothing Then Vendor_RefreshList
5 End Sub
6
7 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
8     If Target.CountLarge > 1 Then Exit Sub
9     If Shapes("EditGrp" ).Visible = True Then Shapes("EditGrp" ).Visible = msoFalse
10    'Hide Edit Group if visible
11    If Not Intersect(Target, Range("D4:K99999" )) Is Nothing And Range("D" & Target.Row
12    ).Value <> Empty Then
13        Range("B3" ).Value = Target.Row 'Set Selected Row
14        Range("B4" ).Value = Range("L" & Target.Row).Value 'Set DB Row
15        With Shapes("EditGrp" )
16            .Left = Range("L" & Target.Row).Left
17            .Top = Range("L" & Target.Row).Top
18            .Visible = msoCTrue
19        End With
20    End If
21 End Sub
```

```
1 Option Explicit
2
3
4 Private Sub CancelBtn_Click()
5     Unload Me
6 End Sub
7
8 Private Sub SaveBtn_Click()
9     Customer_SaveUpdate
10 End Sub
```

```
1  
2  
3 Option Explicit  
4  
5 Private Sub BrowseBtn_Click()  
6     Employee_AddPicture  
7 End Sub  
8  
9 Private Sub CancelBtn_Click()  
10    Unload Me  
11 End Sub  
12  
13 Private Sub Field3_Change()  
14     If Field3.Value = "Hourly" Then  
15         RateLbl.Caption = "Hourly Rate"  
16     Else  
17         RateLbl.Caption = "Annual Amount"  
18     End If  
19 End Sub  
20  
21 Private Sub SaveBtn_Click()  
22     If ActiveSheet.Name = "Employees" Then  
23         EmployeeList_SaveUpdate  
24     Else 'Payroll  
25         EmployeePayroll_SaveUpdate  
26     End If  
27 End Sub
```

```
1  
2 Option Explicit  
3  
4 Private Sub CancelBtn_Click()  
5     Unload Me  
6 End Sub  
7  
8 Private Sub OkBtn_Click()  
9     UserLogin  
10 End Sub  
11
```

```
1 Option Explicit
2
3
4
5 Private Sub EnableCheckboxes_Click()
6     If Me.EnableCheckboxes.Value = True Then
7         Me.TreeView1.CheckBoxes = True
8     Else
9         Me.TreeView1.CheckBoxes = False
10    End If
11 End Sub
12
13 Private Sub IncInvoice_Click()
14     If IncInvoice.Value = True Then OrderDB.Range("AC3:AC5").Value = "INVOICE" Else
15         OrderDB.Range("AC3:AC5").Value = "=" & Chr(34) & "" & Chr(34)
16     CreateTreeView
17 End Sub
18
19 Private Sub IncPurch_Click()
20     If IncPurch.Value = True Then OrderDB.Range("AB3:AB5").Value = "PURCHASE" Else
21         OrderDB.Range("AB3:AB5").Value = "=" & Chr(34) & "" & Chr(34)
22     CreateTreeView
23 End Sub
24
25 Private Sub IncW0_Click()
26     If IncW0.Value = True Then OrderDB.Range("AA3:AA5").Value = "WORK ORDER" Else
27         OrderDB.Range("AA3:AA5").Value = "=" & Chr(34) & "" & Chr(34)
28     CreateTreeView
29 End Sub
30
31
32
33
34 Private Sub PrintBtn_Click()
35     PrintSelectedOrdersOld
36 End Sub
37
38 Private Sub SettingImageList()
39     Dim MyImgList As New ImageList
40     With MyImgList.ListImages
41         Dim ImageNum As Long
42         ImageNum = 1
43         .Add Key:="Invoices", Picture:=OrderDB.Invoice.Picture
44         .Add Key:="Work Orders", Picture:=OrderDB.Work_Order.Picture
45         .Add Key:="Purchases", Picture:=OrderDB.Purchase.Picture
46     End With
47     With TreeView1
48         Set .ImageList = MyImgList
49         .Indentation = 1
50         .LabelEdit = tvwManual
51         .HideSelection = False
52     End With
53 End Sub
54
55
56 Private Sub TreeView1_Click()
57     Dim SelItem As String, PicPath As String
58     SelItem = Me.TreeView1.SelectedItem.Text
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

1

```
1
59   If InStr(SelItem, "#" ) > 0 Then
60     PicPath = [OrderPicFolder] & "\" & Me.TreeView1.SelectedItem.Key & ".jpg"
61     If Dir(PicPath, vbDirectory) <> "" Then Me.OrderPic.Picture = LoadPicture(
        PicPath)
62   End If
63 End Sub
64
65 Private Sub UpdateOrdersBtn_Click()
66   CreateTreeView
67 End Sub
68
69 Private Sub UserForm_Activate()
70   SettingImageList
71   CreateTreeView
72 End Sub
```



```
1 Option Explicit
2
3 Private Sub Checkboxes_Click()
4     If Me.CheckBoxes.Value = True Then
5         Me.TreeView2.CheckBoxes = True
6     Else
7         Me.TreeView2.CheckBoxes = False
8     End If
9 End Sub
10
11
12 Private Sub CommandButton1_Click()
13
14 End Sub
15
16 Private Sub IncInvoices_Click()
17     GenerateTreeView
18 End Sub
19
20 Private Sub IncPurchases_Click()
21     GenerateTreeView
22 End Sub
23
24 Private Sub IncWorkOrders_Click()
25     GenerateTreeView
26 End Sub
27
28 Private Sub PrintBtn_Click()
29     PrintSelectedOrders 'Run Macro To Print Orders
30 End Sub
31
32 Private Sub SettingImages()
33     Dim MyImgList As New ImageList
34     With MyImgList.ListImages
35         Dim ImageNumb As Long
36         ImageNumb = 1
37         .Add Key:="Invoices" , Picture:=OrderDB.Invoice.Picture
38         .Add Key:="Work Orders" , Picture:=OrderDB.Work_Order.Picture
39         .Add Key:="Purchases" , Picture:=OrderDB.Purchase.Picture
40     End With
41     With TreeView2
42         Set .ImageList = MyImgList
43         .Indentation = 1
44         .LabelEdit = tvwManual
45         .HideSelection = False
46     End With
47
48 End Sub
49
50 Private Sub TreeView2_Click()
51     Dim SelItem As String, PicPath As String
52     On Error Resume Next
53     SelItem = Me.TreeView2.SelectedItem.Text
54     If InStr(SelItem, "#" ) > 0 Then
55         PicPath = [OrderPicFolder] & "\" & Me.TreeView2.SelectedItem.Key & ".jpg"
56         If Dir(PicPath, vbDirectory) <> "" Then Me.OrderPic.Picture = LoadPicture(
57             PicPath)
58     End If
59     On Error GoTo 0
60 End Sub
```

```
61 Private Sub UserForm_Activate()  
62     SettingImages  
63     GenerateTreeView  
64 End Sub
```

```
1  
2  
3 Option Explicit  
4  
5 Private Sub BrowseBtn_Click()  
6     Product_AddPicture  
7 End Sub  
8  
9 Private Sub CancelBtn_Click()  
10    Unload Me  
11 End Sub  
12  
13  
14 Private Sub SaveBtn_Click()  
15    Product_SaveUpdate  
16 End Sub
```

```
1  
2 Option Explicit  
3  
4 Private Sub BrowseBtn_Click()  
5     Resource_AddPicture  
6 End Sub  
7  
8 Private Sub CancelBtn_Click()  
9     Unload Me  
10 End Sub  
11  
12 Private Sub ResType_Change()  
13     Dim TypeRow As Long  
14     On Error Resume Next  
15     TypeRow = Admin.Range("ResourceTypes").Find(Me.ResType.Value, , xlValues, xlWhole).  
16     Row  
17     On Error GoTo 0  
18     If TypeRow <> 0 Then Me.LblDefCap.Caption = Admin.Range("C" & TypeRow).Value & "  
19     Capacity" Else Me.LblDefCap.Caption = "Default Capacity"  
20 End Sub  
21  
22 Private Sub SaveBtn_Click()  
23     Resource_SaveUpdate  
24 End Sub
```

```
1
2 Option Explicit
3
4 Private Sub CancelBtn_Click()
5     Unload Me
6 End Sub
7
8
9 Private Sub Field3_Change() 'Add/Update Account Type
10    Dim AcctRow As Long
11    If Field3.Value = Empty Then Exit Sub
12    On Error Resume Next
13    AcctRow = Admin.Range("TransAccts" ).Find(Field3.Value, , xlValues, xlWhole).Row
14    On Error GoTo 0
15    If AcctRow <> 0 Then Field4.Value = Admin.Range("W" & AcctRow).Value 'Set Account
    Type
16 End Sub
17
18 Private Sub SaveBtn_Click()
19     Transaction_SaveUpdate
20 End Sub
```

```
1  
2  
3 Option Explicit  
4  
5 Private Sub BrowseBtn_Click()  
6     User_AddPicture  
7 End Sub  
8  
9 Private Sub CancelBtn_Click()  
10    Unload Me  
11 End Sub  
12  
13  
14 Private Sub SaveBtn_Click()  
15    User_SaveUpdate  
16 End Sub
```

```
1
2
3
4
5 Option Explicit
6
7
8 Private Sub CancelBtn_Click()
9     Unload VendorFrm
10 End Sub
11
12 Private Sub CancelBtn_Exit(ByVal Cancel As MSForms.ReturnBoolean)
13     Unload VendorFrm
14 End Sub
15
16 Private Sub SaveBtn_Click()
17     Vendor_SaveUpdate
18 End Sub
19
20 Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
21     ' CloseMode = 0: User closed the form by clicking the "X" button
22     ' CloseMode = 1: User closed the form by clicking the Close button (if present)
23
24     ' If CloseMode = 0 Then
25     ' Perform any necessary cleanup or actions before closing the form
26
27     ' Unload the UserForm to release resources
28     'Unload VendorFrm
29     'End If
30 End Sub
```

```
1 Option Explicit
2
3 Dim PicFolder As FileDialog
4
5 Sub BrowseForEmployeePicFolder()
6     Set PicFolder = Application.FileDialog(msoFileDialogFolderPicker)
7     With PicFolder
8         .Title = "Browse Product Picture Folder"
9         .AllowMultiSelect = False
10        If .Show <> -1 Then GoTo NoSelection
11        Admin.Range("C4").Value = .SelectedItems(1)
12    End With
13    NoSelection:
14 End Sub
15
16 Sub BrowseForMainSharedFolder()
17     Dim SharedFolder As FileDialog
18     Set SharedFolder = Application.FileDialog(msoFileDialogFolderPicker)
19     With SharedFolder
20         .Title = "Please select the main shared folder within Dropbox as instructed by
21         your Administrator"
22         .AllowMultiSelect = False
23         If .Show <> -1 Then GoTo NotSelected
24         If InStr(.SelectedItems(1), "Dropbox") = 0 Then
25             MsgBox "Please make sure to select a folder within Drobox"
26             Exit Sub
27         End If
28         Admin.Range("C3").Value = .SelectedItems(1) 'Full Folder Path
29     End With
30 End Sub
31
32 Sub BrowseForOrderPicFolder()
33     Set PicFolder = Application.FileDialog(msoFileDialogFolderPicker)
34     With PicFolder
35         .Title = "Browse Order Picture Folder"
36         .AllowMultiSelect = False
37         If .Show <> -1 Then GoTo NoSelection
38         Admin.Range("C7").Value = .SelectedItems(1)
39     End With
40    NoSelection:
41 End Sub
42
43
44 Sub BrowseForProductPicFolder()
45     Set PicFolder = Application.FileDialog(msoFileDialogFolderPicker)
46     With PicFolder
47         .Title = "Browse Product Picture Folder"
48         .AllowMultiSelect = False
49         If .Show <> -1 Then GoTo NoSelection
50         Admin.Range("C5").Value = .SelectedItems(1)
51     End With
52    NoSelection:
53 End Sub
54
55 Sub BrowseForUserPicFolder()
56     Set PicFolder = Application.FileDialog(msoFileDialogFolderPicker)
57     With PicFolder
58         .Title = "Browse User Picture Folder"
59         .AllowMultiSelect = False
60         If .Show <> -1 Then GoTo NoSelection
```



```
61 | 1 2 Admin.Range("C6" ).Value = .SelectedItems(1)
62 | End With
63 | NoSelection:
64 | End Sub
```

```

1 Option Explicit
2 Dim SharedFold As String, Role As String, Username As String
3
4 Sub SharedFolder_Check()
5     'Check For Correct Shared Folder
6     SharedFold = [SharedFolder] 'Set Shared Folder
7     If SharedFold = "" Or Dir(SharedFold, vbDirectory) = "" Then
8         MsgBox "Please select a correct shared folder"
9         BrowseForMainSharedFolder
10        SharedFold = [SharedFolder] 'Set Shared Folder
11        If SharedFold = "" Or Dir(SharedFold, vbDirectory) = "" Or InStr(SharedFold,
12            "Dropbox" ) = 0 Then End
13    End If
14    'Check For User Picture Folder
15    If Dir(SharedFold & "/User Pics" , vbDirectory) = "" Then Mkdir (SharedFold &
16        "/User Pics" ) 'Create User Picture folder if missing
17 End Sub
18
19 Sub Workbook_CheckAndSync()
20     Dim FileName As String, FilePath As String, FileText As String, OldValue As String,
21     TextArr() As String, newValue
22     Dim SheetNm As String, CellAdd As String, CellVal As String, SentFrom As String,
23     Mesg As String, UserFolder As String, CurrentUser As String
24     Dim MsgNumb As Long, ChangeRow As Long, TextFile As Integer
25     Dim TopPos As Double, LeftPos As Double
26     With Admin
27         If [LoggedInUser] = Empty Then Exit Sub 'Exit On New User
28         SharedFolder_Check 'Run Macro to check for Correct shared folder
29         SharedFold = [SharedFolder] 'Set Shared Folder
30         CurrentUser = [LoggedInUser] 'Set User Name
31         If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (SharedFold
32             & "\User Changes" )
33         UserFolder = SharedFold & "\User Changes\" & CurrentUser & "\" 'User Folder
34         If Dir(UserFolder, vbDirectory) = "" Then Mkdir (UserFolder) 'Create Folder if
35             it does not exist
36         MsgNumb = 1 'Set Intial Message Number
37         [ShareSyncRun] = True 'Set File Sync To True
38         FileName = Dir(UserFolder & "*.txt" ) 'Check for any/all text files in folder
39         TextFile = FreeFile
40         Do While Len(FileName) > 0
41             FilePath = UserFolder & FileName
42             Open FilePath For Input As TextFile
43             FileText = Input(LOF(TextFile), TextFile)
44             Close TextFile
45             TextArr = Split(FileText, vbNewLine)
46             Username = TextArr(0) 'User Who made change
47             newValue = TextArr(1) 'Changed (new) Value
48             Close #1
49
50             'All Cell Changes
51             SheetNm = Left(FileName, InStr(FileName, "--" ) - 1) 'Extract Sheet Name
52             CellAdd = Replace(Right(FileName, Len(FileName) - InStr(FileName, "--" ) - 1),
53                 ".txt" , "" ) 'Cell Address
54             If ThisWorkbook.Sheets(SheetNm).Range(CellAdd).Value <> newValue Then 'Only
55                 Update if not the same
56                 ThisWorkbook.Sheets(SheetNm).Range(CellAdd).Value = newValue 'Make Cell
57                 Change
58             End If
59             NextFile:
60             FileName = Dir() 'Clear File Name
61         End While
62     End With
63 End Sub

```

1 2 3

```
53 | 1 2 3 Kill (FilePath) 'Delete File
54 | | | Loop
55 | | | [ShareSyncRun] = False 'Set Syncing to False
56 | | End With
57 | End Sub
58 |
59 |
60 |
61 |
62 |
63 |
```

```

1 Option Explicit
2 Dim CustRow As Long, CustCol As Long, LastRow As Long, LastResultRow As Long
3 Dim CustFld As control
4
5 Sub Customer_AddNew()
6     Customers.Range("B3,B4" ).ClearContents 'Clear Selected Customer Row and Cust DB
7     Row
8     CustomerFrm.Show
9 End Sub
10
11 Sub Customer_ClearFilter()
12     Customers.Range("F2:H2" ).ClearContents
13 End Sub
14
15 Sub Customer_Delete()
16     If MsgBox("Are you sure you want to delete this Customer?" , vbYesNo, "Delete
17     Customer" ) = vbNo Then Exit Sub
18     If Customers.Range("B4" ).Value = Empty Then Exit Sub
19     CustRow = Customers.Range("B4" ).Value 'Customer Row
20     For CustCol = 2 To 9
21         CustomerDB.Cells(CustRow, CustCol).ClearContents
22     Next CustCol
23     Customers.Shapes("EditGrp" ).Visible = msoFalse
24     Customers.Range("B3,B4" ).ClearContents
25     Customer_RefreshList
26 End Sub
27
28
29 Sub Customer_Edit()
30     With Customers
31         If .Range("B4" ).Value = Empty Then
32             MsgBox "Please select on a Customer to edit"
33             Exit Sub
34         End If
35         CustRow = .Range("B4" ).Value 'Customer Row
36         With CustomerFrm
37             For CustCol = 2 To 8
38                 Set CustFld = .Controls("Field" & CustCol - 1)
39                 CustFld.Value = CustomerDB.Cells(CustRow, CustCol).Value 'Add Customer
40                 Data
41             Next CustCol
42             .Show 'Display User Form
43         End With
44     End With
45 End Sub
46
47 Sub Customer_RefreshList()
48     With CustomerDB
49         Customers.Range("D4:L9999" ).ClearContents
50         LastRow = .Range("A9999" ).End(xlUp).Row
51         If LastRow < 4 Then Exit Sub
52         Application.ScreenUpdating = False
53         .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
54         "M2:N3" ), CopyToRange:=.Range("P2:X2" ), Unique:=True
55         LastResultRow = .Range("X99999" ).End(xlUp).Row
56         If LastResultRow < 3 Then
57             Application.ScreenUpdating = True
58             Exit Sub
59         End If
60     End With
61 End Sub

```

1 2 3

```

58 | 1 2 3 | End If
59 | | If LastResultRow < 4 Then GoTo NoSort 'No Sort on a single item
60 | | 'Sort By ID
61 | | With .Sort
62 | |     .SortFields.Clear
63 | |     .SortFields.Add Key:=CustomerDB.Range("P3" ), SortOn:=xlSortOnValues, Order:=
64 | |     xlAscending, DataOption:=xlSortNormal 'Sort
65 | |     .SetRange CustomerDB.Range("P3:X" & LastResultRow) 'Set Range
66 | |     .Apply 'Apply Sort
67 | | End With
68 | | NoSort:
69 | | Customers.Range("D4:L" & LastResultRow + 1).Value = .Range("P3:X" &
70 | | LastResultRow).Value
71 | | Application.ScreenUpdating = True
72 | | End With
73 | | End Sub
74 | Sub Customer_SaveUpdate()
75 |     'Check For Required field
76 |     With CustomerFrm
77 |         If .Field1.Value = Empty Then
78 |             MsgBox "Please make sure to add a Customer Name before saving"
79 |             Exit Sub
80 |         End If
81 |         If Customers.Range("B4" ).Value = Empty Then 'New Customer
82 |             CustRow = CustomerDB.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
83 |             CustomerDB.Range("A" & CustRow).Value = Customers.Range("B5" ).Value 'Next
84 |             Customer ID
85 |             CustomerDB.Range("I" & CustRow).Value = "=Row()"
86 |         Else 'Existing Cust.
87 |             CustRow = Customers.Range("B4" ).Value 'Customer Row
88 |         End If
89 |         For CustCol = 2 To 8
90 |             Set CustFld = .Controls("Field" & CustCol - 1)
91 |             CustomerDB.Cells(CustRow, CustCol).Value = CustFld.Value
92 |         Next CustCol
93 |     End With
94 |     Unload CustomerFrm
95 |     Customer_RefreshList
96 | End Sub

```

```

1 Option Explicit
2 Dim EmplRow As Long, EmplCol As Long, LastRow As Long, LastResultRow As Long
3 Dim EmplFld As control
4
5 Sub Employee_AddNew()
6     If ActiveSheet.Name = "Employees" Then 'Employees Screen
7         Employees.Range("B3,B4" ).ClearContents 'Clear Selected Employee Row and Empl DB
8         Row
9     Else 'Payroll Screen
10        Payroll.Range("B7,B10,N4,P4,P6,N8,P8,M12:O9999,S12:S9999" ).ClearContents 'Sel
11        Selected Employee Pay Details
12        EmployeeFrm.Field13.Value = True 'Active Status set to true/checked
13        EmployeeFrm.Show
14    End If
15    EmployeeFrm.Show
16 End Sub
17
18 Sub Employee_AddPicture()
19     Dim EmplPic As FileDialog
20     Dim PicName As String, PicPath As String, PicFolder As String
21
22     If [EmployeeFolder] = "" Then
23         MsgBox "Please select a folder for Employee pictures"
24         BrowseForEmployeePicFolder
25         If [EmployeeFolder] = "" Then Exit Sub
26     End If
27
28     PicFolder = [EmployeeFolder] 'Set Picture Folder
29     Set EmplPic = Application.FileDialog(msoFileDialogFilePicker)
30     With EmplPic
31         .Title = "Select Employee Picture"
32         .Filters.Add "Select Picture", "*.jpg, *.png, *.jpeg, *.gif", 1
33         If .Show <> -1 Then GoTo NoSelection
34         PicPath = .SelectedItems(1) 'Entire Picture Path
35     End With
36
37     PicName = Dir(PicPath) 'Set Picture Name
38     EmployeeFrm.Field12.Value = PicName 'Set Picture Name
39     'Copy Employee Picture to picture folder
40     On Error Resume Next
41     FileCopy PicPath, PicFolder & "\" & PicName
42     On Error GoTo 0
43     NoSelection:
44 End Sub
45
46 Sub Employee_AddUpdatePicture()
47     Dim PicName As String, PicFolder As String, PicPath As String
48     With Payroll
49         If .Range("B8" ).Value = Empty Then Exit Sub
50         EmplRow = .Range("B8" ).Value 'Employee Row
51         PicFolder = [PictureFolder]
52         PicName = EmployeeDB.Range("M" & EmplRow).Value 'Employee Picture
53         If PicFolder = "" Or PicName = "" Then .Shapes("EmplPic" ).Visible = msoFalse
54         PicPath = PicFolder & "\" & PicName
55         If Dir(PicPath, vbDirectory) = "" Then .Shapes("EmplPic" ).Visible = msoFalse
56         .Shapes("EmplPic" ).Fill.UserPicture PicPath
57         .Shapes("EmplPic" ).Visible = msoCTrue
58     End With
59 End Sub
60
61 Sub Employee_ClearFilter()
62     Employees.Range("F2:H2" ).ClearContents
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

1
60 End Sub
61
62 Sub Employee_Delete()
63     If MsgBox("Are you sure you want to delete this Employee?" , vbYesNo, "Delete
Employee" ) = vbNo Then Exit Sub
64     If Employees.Range("B4" ).Value = Empty Then Exit Sub
65     EmplRow = Employees.Range("B4" ).Value 'Employee Row
66     For EmplCol = 2 To 14
67         EmployeeDB.Cells(EmplRow, EmplCol).ClearContents
68     Next EmplCol
69     Employees.Shapes("EditGrp" ).Visible = msoFalse
70     Employees.Range("B3,B4" ).ClearContents
71     Employee_RefreshList
72 End Sub
73
74
75
76 Sub Employee_Edit()
77     With Employees
78         If .Range("B4" ).Value = Empty Then
79             MsgBox "Please select on a Employee to edit"
80             Exit Sub
81         End If
82         EmplRow = .Range("B4" ).Value 'Employee Row
83         With EmployeeFrm
84             For EmplCol = 2 To 14
85                 Set EmplFld = .Controls("Field" & EmplCol - 1)
86                 EmplFld.Value = EmployeeDB.Cells(EmplRow, EmplCol).Value 'Add Employee
Data
87             Next EmplCol
88             .Show 'Display User Form
89         End With
90     End With
91 End Sub
92
93
94 Sub Employee_RefreshList()
95     With EmployeeDB
96         Employees.Range("D4:Q9999" ).ClearContents
97         LastRow = .Range("A9999" ).End(xlUp).Row
98         If LastRow < 4 Then Exit Sub
99         Application.ScreenUpdating = False
100        .Range("A3:O" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"AB2:AC3" ), CopyToRange:=.Range("AE2:AS2" ), Unique:=True
101        LastResultRow = .Range("AE99999" ).End(xlUp).Row
102        If LastResultRow < 3 Then
103            Application.ScreenUpdating = True
104            Exit Sub
105        End If
106        If LastResultRow < 4 Then GoTo NoSort 'No Sort on a single item
107        'Sort By ID
108        With .Sort
109            .SortFields.Clear
110            .SortFields.Add Key:=EmployeeDB.Range("AE3" ), SortOn:=xlSortOnValues, Order:=
xlAscending, DataOption:=xlSortNormal 'Sort
111            .SetRange EmployeeDB.Range("AE3:AS" & LastResultRow) 'Set Range
112            .Apply 'Apply Sort
113        End With
114        NoSort:
115        Employees.Range("D4:R" & LastResultRow + 1).Value = .Range("AE3:AS" &
LastResultRow).Value

```

1 2

```

1 2
116     Employees.Range("F2" ).Select
117     Application.ScreenUpdating = True
118     End With
119     End Sub
120 Sub EmployeeList_SaveUpdate()
121     'Check For Required field
122     With EmployeeFrm
123     If .Field1.Value = Empty Then
124         MsgBox "Please make sure to add a Employee Name before saving"
125         Exit Sub
126     End If
127     If Employees.Range("B4" ).Value = Empty Then 'New Employee
128         EmplRow = EmployeeDB.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
129         EmployeeDB.Range("A" & EmplRow).Value = Employees.Range("B5" ).Value 'Next
130         Employee ID
131         EmployeeDB.Range("O" & EmplRow).Value = "=Row()"
132     Else 'Existing Empl.
133         EmplRow = Employees.Range("B4" ).Value 'Employee Row
134     End If
135     For EmplCol = 2 To 14
136         Set EmplFld = .Controls("Field" & EmplCol - 1)
137         EmployeeDB.Cells(EmplRow, EmplCol).Value = EmplFld.Value
138     Next EmplCol
139     End With
140     Unload EmployeeFrm
141     Employee_RefreshList
142 End Sub
143
144 Sub EmployeePayroll_SaveUpdate()
145     'Check For Required field
146     Dim EmpType As String, TypeRow As Long
147     With EmployeeFrm
148     If .Field1.Value = "" Or .Field2.Value = Empty Then
149         MsgBox "Please make sure to have both a Employee Name and Employee Type, as
150         they are required"
151         Exit Sub
152     End If
153     If Payroll.Range("B8" ).Value = Empty Then 'New Employee
154         EmplRow = Employees.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
155         Payroll.Range("B7" ).Value = Payroll.Range("B9" ).Value 'Next Employee ID
156         EmployeeDB.Range("A" & EmplRow).Value = Payroll.Range("B7" ).Value
157         EmployeeDB.Range("O" & EmplRow).Value = "=Row()"
158     Else 'Existing Emp.
159         EmplRow = Payroll.Range("B8" ).Value 'Employee Row
160     End If
161     For EmplCol = 2 To 14
162         Set EmplFld = .Controls("Field" & EmplCol - 1)
163         EmployeeDB.Cells(EmplRow, EmplCol).Value = EmplFld.Value
164     Next EmplCol
165     'Update Payroll Fields
166     Payroll.Range("N4" ).Value = .Field1.Value 'Employee Name
167     Payroll.Range("P4" ).Value = .Field3.Value 'Payroll Type
168     If Payroll.Range("P4" ).Value <> "Hourly" Then
169         On Error Resume Next
170         Payroll.Range("N8" ).Value = (.Field4.Value / Payroll.Range("B11" ).Value)
171         'Period Salary
172         On Error GoTo 0
173     Else 'Standard hourly
174         Payroll.Range("N8" ).Value = .Field4.Value 'Hourly Regular Rate
175         Payroll.Range("P8" ).Value = .Field5.Value 'Hourly OT Rate

```

1 2 3


```
174 | 1 2 3  
    | | |  
    | | End If  
175 | |  
    | | End With  
176 | |  
177 | | EmployeeFrm.Hide 'Hide Form  
178 | | Unload EmployeeFrm  
179 | | Employee_AddUpdatePicture  
180 | End Sub
```

```
1 Option Explicit
```

```
4 Sub ChangeLanguage()
```

```
5 Dim LangNumb As Long, LangRow As Long, LangCol As Long, LastRow As Long
```

```
6 Dim ButShp As Shape
```

```
7 Dim FrmName As String
```

```
8 Dim WkSht As Worksheet
```

```
9 Dim AppUserFrm As Object
```

```
10 Dim LabCont As control
```

```
11 Dim ShtCdNm As String, SellLang As String
```

```
12 With LangDB
```

```
13 SellLang = [SellLanguage]
```

```
14 On Error Resume Next
```

```
15 LangCol = .Range("C3:Z3").Find(SellLang, , xlValues, xlWhole).Column
```

```
16 On Error GoTo 0
```

```
17 If LangCol = 0 Then
```

```
18 MsgBox "Please select a correct language"
```

```
19 Exit Sub
```

```
20 End If
```

```
21 Application.ScreenUpdating = False
```

```
22 Application.EnableEvents = False
```

```
23 Application.Calculation = xlCalculationManual
```

```
24 'Update Sheet Fields
```

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

```
26 If LastRow < 4 Then GoTo NoSheetFields
```

```
27 For LangRow = 4 To LastRow
```

```
28 If .Range("A" & LangRow).Value <> ShtCdNm Then
```

```
29 Set WkSht = Nothing
```

```
30 ShtCdNm = .Range("A" & LangRow).Value 'Set Sheet To Variable
```

```
31 For Each WkSht In ThisWorkbook.Sheets
```

```
32 If WkSht.CodeName = ShtCdNm Then Exit For
```

```
33 Next WkSht
```

```
34 End If
```

```
35 WkSht.Range(.Range("B" & LangRow).Value).Value = .Cells(LangRow, LangCol).Value 'Set Field Language
```

```
36 Next LangRow
```

```
37 NoSheetFields:
```

```
38 'Change Buttons
```

```
40 LastRow = .Range("AA99999").End(xlUp).Row
```

```
41 If LastRow < 4 Then GoTo NoButtons
```

```
42 For LangRow = 4 To LastRow
```

```
43 If .Range("AA" & LangRow).Value <> ShtCdNm Then
```

```
44 Set WkSht = Nothing
```

```
45 ShtCdNm = .Range("AA" & LangRow).Value
```

```
46 For Each WkSht In ThisWorkbook.Sheets
```

```
47 If WkSht.CodeName = ShtCdNm Then Exit For
```

```
48 Next WkSht
```

```
49 End If
```

```
50 WkSht.Shapes(.Range("AB" & LangRow).Value).TextFrame2.TextRange.Text = .Cells(LangRow, LangCol + 26).Value 'Set Button Language
```

```
51 Next LangRow
```

```
52 NoButtons:
```

```
53 'Change Userform Labels
```

```
55 LastRow = .Range("BA99999").End(xlUp).Row
```

```
56 RestartFrms:
```

```
57 If LastRow < 4 Then GoTo NoLabels
```

```
58 'On Error Resume Next
```

```
59 For LangRow = 4 To LastRow
```

```
1 2 3
```

```

60      1 2 3      Set AppUserFrm = ThisWorkbook.VBProject.VBComponents(.Range("BA" & LangRow).
        Value).Designer 'Set Userform
61      'On Error Resume Next
62      Set LabCont = Nothing
63      If AppUserFrm Is Nothing Then 'Manually set forms when form has been opened
64          FrmName = .Range("BA" & LangRow).Value 'Set Form Name
65          Select Case FrmName
66              Case "CustomerFrm"
67                  Set AppUserFrm = CustomerFrm
68              Case "EmployeeFrm"
69                  Set AppUserFrm = EmployeeFrm
70              Case "LoginFrm"
71                  Set AppUserFrm = LoginFrm
72              Case "OrdersFrm"
73                  Set AppUserFrm = OrdersFrm
74              Case "OrdersTreeView"
75                  Set AppUserFrm = OrdersTreeView
76              Case "ProductFrm"
77                  Set AppUserFrm = ProductFrm
78              Case "ResourceFrm"
79                  Set AppUserFrm = ResourceFrm
80              Case "TransactFrm"
81                  Set AppUserFrm = TransactFrm
82              Case "UserFrm"
83                  Set AppUserFrm = UserFrm
84              Case "VendorFrm"
85                  Set AppUserFrm = VendorFrm
86          End Select
87      End If
88      Set LabCont = AppUserFrm.Controls(.Range("BB" & LangRow).Value)
89      LabCont.Caption = .Cells(LangRow, LangCol + 52).Value 'Set Label Caption
90  Next LangRow
91      Set LabCont = Nothing
92      Set AppUserFrm = Nothing
93      NoLabels:
94
95
96      'Change Userform Buttons
97      LastRow = .Range("CA99999" ).End(xlUp).Row
98      If LastRow < 4 Then GoTo NoUserformButtons
99      For LangRow = 4 To LastRow
100         Set AppUserFrm = ThisWorkbook.VBProject.VBComponents(.Range("CA" & LangRow).
            Value).Designer 'Set Userform
101         If AppUserFrm Is Nothing Then 'Manually set forms when form has been opened
102             FrmName = .Range("CA" & LangRow).Value 'Set Form Name
103             Select Case FrmName
104                 Case "CustomerFrm"
105                     Set AppUserFrm = CustomerFrm
106                 Case "EmployeeFrm"
107                     Set AppUserFrm = EmployeeFrm
108                 Case "LoginFrm"
109                     Set AppUserFrm = LoginFrm
110                 Case "OrdersFrm"
111                     Set AppUserFrm = OrdersFrm
112                 Case "OrdersTreeView"
113                     Set AppUserFrm = OrdersTreeView
114                 Case "ProductFrm"
115                     Set AppUserFrm = ProductFrm
116                 Case "ResourceFrm"
117                     Set AppUserFrm = ResourceFrm
118                 Case "TransactFrm"

```

1 2 3 4 5

```

119     Set AppUserFrm = TransactFrm
120     Case "UserFrm"
121         Set AppUserFrm = UserFrm
122     Case "VendorFrm"
123         Set AppUserFrm = VendorFrm
124     End Select
125 End If
126 Set LabCont = AppUserFrm.Controls(.Range("CB" & LangRow).Value)
127 LabCont.Caption = .Cells(LangRow, LangCol + 78).Value 'Set Button Caption
128 Next LangRow
129 NoUserformButtons:
130 Set LabCont = Nothing
131 Set AppUserFrm = Nothing
132
133 Application.EnableEvents = True
134 Application.ScreenUpdating = True
135 Application.Calculation = xlCalculationAutomatic
136 End With
137 End Sub

```

```

141 Sub GetButtonText()
142     Dim LangRow As Long
143     Dim ButShp As Shape
144     With ActiveSheet
145         'Get Any Button Texts
146         LangRow = LangDB.Range("AA99999" ).End(xlUp).Row + 1
147         For Each ButShp In .Shapes
148             If ButShp.Type = 13 Or ButShp.Type = 6 Or ButShp.Type = 28 Then GoTo NextShp
149             'Skip Pictures, Groups & Icons
150             On Error GoTo NextShp 'pIctures do not have text
151             If ButShp.TextFrame2.TextRange.Text <> Empty Then
152                 LangDB.Range("AA" & LangRow).Value = .CodeName 'Set Sheet Code Name
153                 LangDB.Range("AB" & LangRow).Value = ButShp.Name 'Button Name
154                 LangDB.Range("AC" & LangRow).Value = ButShp.TextFrame2.TextRange.Text
155                 'Button Text
156                 LangRow = LangRow + 1
157             End If
158         NextShp:
159     Next ButShp
160 End With
161 End Sub

```

```

163 'Extract Worksheet Label Values
164 Sub GetLabelValues()
165     Dim LangRow As Long
166     Dim Cell As Range
167     LangRow = LangDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail. Row
168     With ActiveSheet
169         For Each Cell In .UsedRange
170             If Cell.Interior.Color <> RGB(255, 255, 255) And Not Cell.HasFormula And Cell.
171                 Value <> "" Then
172                 LangDB.Range("A" & LangRow).Value = .CodeName 'Set Sheet Code Name
173                 LangDB.Range("B" & LangRow).Value = Cell.Address 'Add Cell Address
174                 LangDB.Range("C" & LangRow).Value = Cell.Value 'Cell Value
175                 LangRow = LangRow + 1
176             End If
177         Next Cell

```

```

177     End With
178 End Sub
179
180
181 Sub GetUserformLabelsAndButtons()
182     Dim LangRow As Long, ButRow As Long
183     Dim AppUserFrm As Object
184     Dim LabCont As control
185     LangRow = 4
186     ButRow = 4
187     For Each AppUserFrm In ThisWorkbook.VBProject.VBComponents
188         If AppUserFrm.Type = 3 Then 'Userform Type Objects
189             For Each LabCont In AppUserFrm.Designer.Controls
190                 'For Userform Labels
191                 If TypeOf LabCont Is MSForms.Label Then 'Only On Label Controls
192                     LangDB.Range("BA" & LangRow).Value = AppUserFrm.Name 'Set Userform
193                     Name
194                     LangDB.Range("BB" & LangRow).Value = LabCont.Name 'Label Name
195                     LangDB.Range("BC" & LangRow).Value = LabCont.Caption 'Label Caption
196                     LangRow = LangRow + 1
197                 End If
198
199                 'For Useform Buttons
200                 If TypeOf LabCont Is MSForms.CommandButton Then
201                     LangDB.Range("CA" & ButRow).Value = AppUserFrm.Name 'Set Sheet Code
202                     Name
203                     LangDB.Range("CB" & ButRow).Value = LabCont.Name 'Button Name
204                     LangDB.Range("CC" & ButRow).Value = LabCont.Caption 'Button Caption
205                     ButRow = ButRow + 1
206                 End If
207             Next LabCont
208         End If
209     Next AppUserFrm
210 End Sub
211
212 Sub Reset()
213     Application.EnableEvents = True
214     Application.ScreenUpdating = True
215     Application.Calculation = xlCalculationAutomatic
216 End Sub
217
218 Sub ShowVendForm()
219     VendorFrm.Show
220 End Sub
221
222 Public Function Translate(WordToTranslate As String, Language As String) As String
223     Dim AIBody As String, ParsedAnswer As String, APIKey As String, Temp As String,
224     Model As String
225     Dim Prompt As String, Response As String
226     Dim StartString As Long, StringLength As Long
227     Dim oXMLHTTP As Object
228     If WordToTranslate = "" Or Language = "" Then Exit Function
229     APIKey = [ChatGPTAPI] 'Get API Key
230     Temp = "0.2"
231     Model = "gpt-3.5-turbo-1106" 'Set Default Model
232     Set oXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP")
233     Prompt = "Translate only the word(s) " & "\" & Chr(34) & WordToTranslate & "\" &
234     Chr(34) & " into " & Language
235     'Debug.Print Prompt
236     oXMLHTTP.Open "POST", "https://api.openai.com/v1/chat/completions", False 'Set

```

```

1
Open Posts For Text
234 AIBody = "{" "model" ": " "" & Model & "" ", " "messages" ": [{" "role" ": "
      "user" ", " "content" ": " "" & Prompt & "" "}], " "temperature" ": " & Temp & ",
      " "max_tokens" ": 3500}" 'Request Body For Text Response
235 oXMLHTTP.setRequestHeader "Content-Type" , "application/json" 'Set Header Type
236 oXMLHTTP.setRequestHeader "Authorization" , "Bearer " & APIKey 'Set Header
      Authorization
237 'Debug.Print AIBody
238 oXMLHTTP.Send AIBody
239 Response = oXMLHTTP.responseText 'Extract Response
240 Debug.Print Response
241 If InStr(Response, "You exceeded your current quota" ) > 0 Then 'This will appear
      when quota is reached
242     MsgBox "Your current quitot of toekns has been exceeded"
243     End
244     Exit Function
245 End If
246 If InStr(Response, "!DOCTYPE html" ) > 0 Then 'This will appear when there is a
      site issue or an API issue
247     MsgBox "The OpenAI Server appears to be having issues. Please try again a bit
      later"
248     End
249     Exit Function
250 End If
251 If InStr(Response, "You didn't provide an API key" ) > 0 Then
252     MsgBox "No API Key was added. Please add your API key in the Admin screen. You
      can find your API key at " & vbCrLf &
      "https://platform.openai.com/account/api-keys."
253     Exit Function
254 End If
255 If InStr(Response, "Incorrect API key provided" ) > 0 Then
256     MsgBox "An incorrect API key was provided. Please check your API key or generate
      a new one at " & vbCrLf & "https://platform.openai.com/account/api-keys."
257     Exit Function
258 End If
259 If InStr(Response, "maximum context length" ) > 0 Then
260     MsgBox "You have reached the maximum content length for this input"
261     Exit Function
262 End If
263 'DoEvents
264 ' StartString = InStr(Response, "total_tokens") + 15
265 ' StringLength = (InStr(Response, Chr(10) & Chr(32) & Chr(32) & Chr(125)) -
      (InStr(Response, "total_tokens") + 15))
266 ParsedAnswer = Replace(Response, "\n\n" , vbCrLf) 'Remove any new lines
267 ParsedAnswer = Right(ParsedAnswer, Len(ParsedAnswer) - InStrRev(ParsedAnswer, Chr(34
      ) & "content" & Chr(34) & ":" ) - 10)
268 ParsedAnswer = Left(ParsedAnswer, InStr(ParsedAnswer, "}," ) + 1)
269 ParsedAnswer = Left(ParsedAnswer, Len(ParsedAnswer) - 10)
270 ParsedAnswer = Trim(Replace(Replace(ParsedAnswer, "\n" , Chr(10)), "\" & Chr(34),
      Chr(34)))
271 If Left(ParsedAnswer, 1) = Chr(34) Then ParsedAnswer = Right(ParsedAnswer, Len(
      ParsedAnswer) - 1) 'Remove first Quote (if any)
272 ParsedAnswer = Replace(ParsedAnswer, Chr(34), "" ) 'Remove any remaining quotation
      marks
273 Translate = ParsedAnswer
274 DoEvents
275 End Function

```

```

1 Option Explicit
2 Dim UserRow As Long
3 Dim Username As String, UserPW As String, PicName As String, PicFolder As String,
  PicPath As String
4 Dim GrpShp As Shape
5 Sub UserLogin()
6     With LoginFrm
7         Username = .Username
8         UserPW = .Password
9         If Username = Empty Or UserPW = Empty Then
10            MsgBox "Please make sure to enter a Username and Password"
11            Exit Sub
12        End If
13        On Error Resume Next
14        UserRow = UserDB.Range("User_Name").Find(Username, , xlValues, xlWhole).Row
15        On Error GoTo 0
16        If UserRow = 0 Then
17            MsgBox "Please enter a correct user name"
18            Exit Sub
19        End If
20        If UserDB.Range("C" & UserRow).Value <> UserPW Then
21            MsgBox "Password has not been entered correctly. Please try again"
22            Exit Sub
23        End If
24        Unload LoginFrm
25    End With
26
27    'User Logged in Successfully
28    With StoreMgmt
29        [LoggedInUser] = Username 'Set Username
30        .Range("B22").Value = UserDB.Range("D" & UserRow).Value 'Set User Type
31        PicName = UserDB.Range("E" & UserRow).Value 'User Picture
32        PicFolder = [UserFolder]
33        PicPath = PicFolder & "\" & PicName 'User Picture Path
34        If PicName = Empty Or PicFolder = Empty Then GoTo NoPic
35        If Dir(PicPath, vbDirectory) = "" Then GoTo NoPic
36        .Shapes("Header").GroupItems("UserPic").Fill.UserPicture PicPath 'Add User
  Picture
37        NoPic:
38        .Range("B23").Value = Now 'Set Logged In Date/Time
39        .Shapes("Header").Visible = msoCTrue 'Unhide Header
40        .Shapes("Menu").Visible = msoCTrue 'Unhide Menu
41        .Range("C:D").EntireColumn.Hidden = False 'Unhide Column C
42        MainMenuSelect 'Run Macro To Load Sales
43        ActiveWindow.ScrollColumn = 1
44    End With
45 End Sub
46
47 Sub UserLogin_Show()
48     LoginFrm.Show
49 End Sub
50
51 Sub UserLogout()
52     With StoreMgmt
53         .Range("B21:B23").ClearContents 'Clear User Info
54         .Range("C:BZ").EntireColumn.Hidden = True
55         .Shapes("Menu").Visible = msoFalse
56         For Each GrpShp In .Shapes
57             If InStr(GrpShp.Name, "Grp") Then GrpShp.Visible = msoFalse 'Hide all group
  shapes
58         Next GrpShp

```

```
59 | 1 2 .Shapes("Header" ).Visible = msoFalse  
60 |     .Shapes("StartGrp" ).Visible = msoCTrue  
61 |     End With  
62 | End Sub
```



```

1 Option Explicit
2 Dim OrderRow As Long, OrderCol As Long, ItemRow As Long, LastRow As Long, LastItemRow
3 As Long, ItemDBRow As Long, ItemDBCol As Long
4 Dim LastResultRow As Long, ResultRow As Long, SelRow As Long
5 Dim OrderType As String, NameType As String
6
7 Sub CreateAllOrderImages()
8     LastRow = OrderDB.Range("A9999").End(xlUp).Row
9     With Orders
10        For OrderRow = 4 To LastRow
11            .Range("Q5").Value = OrderDB.Range("A" & OrderRow).Value
12            CreateOrderPicture
13        Next OrderRow
14    End With
15 End Sub
16
17
18 Sub CreateOrderPicture()
19     Dim TempChart As Chart, OrderSht As Worksheet, TempShp As Shape
20     Dim FileName As String
21     Set OrderSht = ThisWorkbook.Sheets("Orders") 'Create Temporary Chart
22     With Orders
23         .Activate
24         .Range("J5:N53").Select
25         Selection.Copy
26         .Pictures.Paste.Name = "TempPic"
27         Set TempShp = OrderSht.Shapes("TempPic")
28         Set TempChart = Charts.Add
29         TempChart.ChartArea.Clear
30         Set TempChart = TempChart.Location(Where:=xlLocationAsObject, Name:=.Name)
31         TempChart.ChartArea.Width = TempShp.Width
32         TempChart.ChartArea.Height = TempShp.Height
33         .Range("Q5").Select
34         TempShp.Copy
35         TempChart.ChartArea.Select
36         TempChart.Paste 'Paste range as image to chart
37         'Save chart image to file
38         FileName = [OrderPicFolder] & "\Order" & OrderSht.Range("B3").Value & ".jpg"
39         TempChart.Export FileName:=FileName, FilterName:="jpg"
40
41         'Clean up
42         .Range("Q5").Activate
43         .ChartObjects(OrderSht.ChartObjects.Count).Delete
44         TempShp.Delete
45     End With
46 End Sub
47
48 Sub Order_AddNew()
49     With Orders
50         .Range("B3,B8,K6:K8,N8,Q5,Q6,I11:M48,O11:O48").ClearContents
51         .Range("B3").Value = .Range("B5").Value 'Next Order ID
52         .Range("N7").Value = Now 'Set Date
53         If .Range("B2").Value = "INVOICE" Then 'Invoice Order
54             .Range("Q7").Value = Admin.Range("S19").Value 'Default Sales Income Account
55             .Range("N8").Value = Admin.Range("S25").Value 'Default Invoice Status
56         ElseIf .Range("B2").Value = "PURCHASE" Then 'Purchase Order
57             .Range("Q7").Value = Admin.Range("S21").Value 'Default Purchase Expense
58             Account
59             .Range("N8").Value = Admin.Range("S27").Value 'Default PO Status
60         End If
61     End With
62 End Sub

```

1 2 3

```

1 2 3
59     Else ' Work Order
60         .Range("Q7" ).ClearContents 'Clear Account Field
61         .Range("N8" ).Value = Admin.Range("S29" ).Value 'Default WO Status
62     End If
63     On Error Resume Next
64     .Range("K6" ).Select
65     On Error GoTo 0
66 End With
67 End Sub

68
69
70 Sub Order_Delete()
71     With Orders
72         If MsgBox("Are you sure you want to delete this order?" , vbYesNo, "Delete
73             Order" ) = vbNo Then Exit Sub
74         If .Range("B4" ).Value = Empty Then GoTo NotSaved
75         OrderRow = .Range("B4" ).Value
76         Application.ScreenUpdating = False
77         For OrderCol = 2 To 8
78             OrderDB.Cells(OrderRow, OrderCol).ClearContents
79         Next OrderCol
80
81         'Delete Order Items
82         With OrderItems
83             LastRow = .Range("A99999" ).End(xlUp).Row
84             If LastRow < 4 Then Exit Sub
85             .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
86                 "N2:O3" ), CopyToRange:=.Range("S2:Y2" ), Unique:=True
87             LastResultRow = .Range("X99999" ).End(xlUp).Row
88             If LastResultRow < 3 Then
89                 Application.ScreenUpdating = True
90                 Exit Sub
91             End If
92             For ResultRow = 3 To LastResultRow
93                 ItemDBRow = .Range("X" & ResultRow).Value 'Item DB Row
94                 For ItemDBCol = 2 To 11
95                     .Cells(ItemDBRow, ItemDBCol).ClearContents 'Clear Contents
96                 Next ItemDBCol
97             Next ResultRow
98         End With
99         NotSaved:
100        Application.ScreenUpdating = False
101        Orders_List 'Reload List
102        Order_AddNew
103    End With
104 End Sub

105 Sub Order_Load()
106     With Orders
107         .Range("B1" ).Value = True 'Set Order Load to True
108         .Range("K6:K8,N8,Q5,Q6,I11:M48,O11:O48" ).ClearContents
109         If .Range("B4" ).Value = Empty Then
110             MsgBox "Please enter a correct Order # to load"
111             .Range("B1" ).Value = False 'Set Order Load to False
112             Exit Sub
113         End If
114         OrderRow = .Range("B4" ).Value
115         OrderType = OrderDB.Range("B" & OrderRow).Value
116
117         If OrderType = "PURCHASE" And .Range("B2" ).Value <> "PURCHASE" Then

```

```

1 2
OrderTab_PO 'Update Order Type If needed
118 If OrderType = "INVOICE" And .Range("B2").Value <> "INVOICE" Then
OrderTab_Invoice 'Update Order Type If needed
119 If OrderType = "WORK ORDER" And .Range("B2").Value <> "WORK ORDER" Then
OrderTab_WO 'Update Order Type If needed
120 Application.ScreenUpdating = False
121 For OrderCol = 2 To 7 '(Exclude total which is calculated by a formula)
122     .Range(OrderDB.Cells(1, OrderCol).Value).Value = OrderDB.Cells(OrderRow,
        OrderCol).Value 'Bring over order data
123 Next OrderCol
124 End With
125 'Load Items
126 With OrderItems
127     LastRow = .Range("A9999").End(xlUp).Row
128     If LastRow < 4 Then Exit Sub
129     .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
        "N2:O3"), CopyToRange:=.Range("S2:Y2"), Unique:=True
130     LastResultRow = .Range("X9999").End(xlUp).Row
131     If LastResultRow < 3 Then
132         Application.ScreenUpdating = True
133         Orders.Range("B1").Value = False 'Set Order Load to False
134         Exit Sub
135     End If
136     For ResultRow = 3 To LastResultRow
137         ItemRow = .Range("Y" & ResultRow).Value 'Item Row
138         Orders.Range("I" & ItemRow & ":M" & ItemRow).Value = .Range("S" & ResultRow
            & ":W" & ResultRow).Value 'Product ID, Name, Qty & Amount
139         Orders.Range("O" & ItemRow).Value = .Range("X" & ResultRow).Value 'DB Row
140     Next ResultRow
141 End With
142 Application.ScreenUpdating = True
143 Orders.Range("B1").Value = False 'Set Order Load to False
144 Orders.Activate
145 Orders.Range("K6").Select
146 End Sub
147
148 Sub Order_Print()
149     Orders.PrintOut , , , , True, , , False
150 End Sub
151
152 Sub Order_SavedMsg()
153     With Orders.Shapes("OrderSavedMsg")
154         Dim i As Long, Delay As Double, StartTime As Double
155         .Visible = msoCTrue
156         For i = 1 To 75
157             .Fill.Transparency = i / 75
158             Delay = 0.005
159             StartTime = Timer
160             Do
161                 DoEvents
162             Loop While Timer - StartTime < Delay
163         Next i
164         .Visible = msoFalse
165     End With
166 End Sub
167 Sub Order_SaveUpdate()
168     With Orders
169         LastItemRow = .Range("J48").End(xlUp).Row
170         If LastItemRow < 11 Then
171             MsgBox "Please make sure to enter at least one product before saving"
172             Exit Sub

```

```

173     End If
174     Application.ScreenUpdating = False
175     If .Range("B4" ).Value = Empty Then 'New Order
176         OrderRow = OrderDB.Range("A99999" ).End(xlUp).Row + 1
177         .Range("B3" ).Value = .Range("B5" ).Value 'Next Order ID
178         OrderDB.Range("A" & OrderRow).Value = .Range("B5" ).Value 'Next Order ID
179     Else 'Existing Order
180         OrderRow = .Range("B4" ).Value 'Existing Order DB Row
181     End If
182     For OrderCol = 2 To 8 'Include Order total
183         OrderDB.Cells(OrderRow, OrderCol).Value = .Range(OrderDB.Cells(1, OrderCol).
184             Value).Value 'Save order data
185     Next OrderCol
186     'Save/Udpate Items
187     For ItemRow = 11 To LastItemRow
188         If .Range("O" & ItemRow).Value = Empty Then 'New Item DB
189             ItemDBRow = OrderItems.Range("A9999" ).End(xlUp).Row + 1 'First avail Row
190             OrderItems.Range("A" & ItemDBRow).Value = .Range("B3" ).Value 'Order #
191             OrderItems.Range("B" & ItemDBRow).Value = .Range("B2" ).Value 'Set Order
192             Type
193             OrderItems.Range("J" & ItemDBRow).Value = ItemRow 'Set Item Row
194             OrderItems.Range("K" & ItemDBRow).Value = "=Row()" 'Database row formula
195             .Range("O" & ItemRow).Value = ItemDBRow
196         Else 'Existing
197             ItemDBRow = .Range("O" & ItemRow).Value 'Existing Order Items DB Row
198             OrderItems.Range("C" & ItemDBRow).Value = .Range("I" & ItemRow).Value
199             'Product ID
200             OrderItems.Range("D" & ItemDBRow).Value = .Range("J" & ItemRow).Value
201             'Product Name
202             OrderItems.Range("E" & ItemDBRow).Value = .Range("N7" ).Value 'Order Date
203             OrderItems.Range("F" & ItemDBRow).Value = .Range("K" & ItemRow).Value
204             OrderItems.Range("G" & ItemDBRow).Value = .Range("L" & ItemRow).Value 'Item
205             Qty
206             OrderItems.Range("H" & ItemDBRow).Value = .Range("M" & ItemRow).Value
207             'Price
208             OrderItems.Range("I" & ItemDBRow).Value = .Range("N" & ItemRow).Value
209             'Total
210         Next ItemRow
211     NoItems:
212     Orders_List 'Reload List
213     Application.ScreenUpdating = True
214     Order_SavedMsg
215 End With
216 End Sub
217
218 Sub Orders_List()
219     Orders.Range("B8,D6:H999" ).ClearContents
220     With OrderDB
221         LastRow = .Range("A99999" ).End(xlUp).Row
222         If LastRow < 4 Then Exit Sub
223         .Range("A3:H" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
224             "M2:M3" ), CopyToRange:=.Range("O2:S2" ), Unique:=True
225         LastResultRow = .Range("O99999" ).End(xlUp).Row
226         If LastResultRow < 3 Then Exit Sub

```

```

1 2
224 If LastResultRow < 4 Then GoTo SkipSort
225 With .Sort
226     .SortFields.Clear
227     .SortFields.Add Key:=OrderDB.Range("P3" ), SortOn:=xlSortOnValues, Order:=
xlDescending, DataOption:=xlSortNormal 'Sort
228     .SetRange OrderDB.Range("O3:S" & LastResultRow) 'Set Range
229     .Apply 'Apply Sort
230 End With
231 SkipSort:
232 Orders.Range("D6:H" & LastResultRow + 3).Value = .Range("O3:S" & LastResultRow)
.Value
233 End With
234 With Orders
235
236     'Set Selected Row (if any)
237     On Error Resume Next
238     SelRow = .Range("D6:D9999" ).Find(.Range("B3" ).Value, , xlFormulas, xlWhole).Row
239
240     If SelRow <> 0 And .Range("B3" ).Value <> Empty Then .Range("B8" ).Value = SelRow
        'Set Selected Order Row
241     On Error GoTo 0
242 End With
243 End Sub
244
245 Sub OrderTab_Invoice()
246 With Orders
247     .Activate
248     With .Shapes("InvTab" ) 'Update Invoice Tab
249         .Fill.ForeColor.RGB = RGB(198, 229, 243) 'Light Blue Background
                .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(47, 85, 151) 'Font Color
                Dark Blue
250     End With
251     With .Shapes("POTab" ) 'Update Invoice Tab
252         .Fill.ForeColor.RGB = RGB(47, 85, 151) 'Dark Blue Background
                .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255) 'White
                Font
253     End With
254     With .Shapes("WOTab" ) 'Update Work Order Tab
255         .Fill.ForeColor.RGB = RGB(47, 85, 151) 'Dark Blue Background
                .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255) 'White
                Font
256     End With
257     .Range("B2" ).Value = "INVOICE"
258
259     'Update Customer Name Validation
260     .Range("K6" ).Select
261     .Range("K6" ).Validation.Delete
262     .Range("K6" ).Validation.Add xlValidateList, , , "=Cust_Name"
263     Orders_List 'Update List of Orders to Invoice Orders
264 End With
265 End Sub
266
267
268
269
270 Sub OrderTab_PO()
271 With Orders
272     .Activate
273     With .Shapes("POTab" ) 'Update PO Tab
274         .Fill.ForeColor.RGB = RGB(198, 229, 243) 'Light Blue Background
                .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(47, 85, 151)
275     End With
276     With Orders.Shapes("InvTab" ) 'Update Invoice Tab
277

```

1 2 3

```
278     .Fill.ForeColor.RGB = RGB(47, 85, 151) 'Dark Blue Background
279     .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255) 'White
280     Font
281     End With
282     With Orders.Shapes("WOTab" ) 'Update Work Order Tab
283         .Fill.ForeColor.RGB = RGB(47, 85, 151) 'Dark Blue Background
284         .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255) 'White
285         Font
286     End With
287     Orders.Range("B2" ).Value = "PURCHASE"
288     'Update Vendor Name Validation
289     .Range("K6" ).Select
290     .Range("K6" ).Validation.Delete
291     .Range("K6" ).Validation.Add xlValidateList, , , "=Vend_Name"
292     End With
293     Orders_List 'Update List of Orders to PO Orders
294 End Sub
295 Sub OrderTab_WO()
296     With Orders
297         .Activate
298         With .Shapes("WOTab" ) 'Update WO Tab
299             .Fill.ForeColor.RGB = RGB(198, 229, 243) 'Light Blue Background
300             .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(47, 85, 151)
301         End With
302         With Orders.Shapes("InvTab" ) 'Update Invoice Tab
303             .Fill.ForeColor.RGB = RGB(47, 85, 151) 'Dark Blue Background
304             .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255) 'White
305             Font
306         End With
307         With Orders.Shapes("POTab" ) 'Update Purchase Order Tab
308             .Fill.ForeColor.RGB = RGB(47, 85, 151) 'Dark Blue Background
309             .TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255) 'White
310             Font
311         End With
312         Orders.Range("B2" ).Value = "WORK ORDER"
313         'Update Customer Name Validation
314         .Range("K6" ).Select
315         .Range("K6" ).Validation.Delete
316         .Range("K6" ).Validation.Add xlValidateList, , , "=Cust_Name"
317     End With
318     Orders_List 'Update List of Orders to PO Orders
319 End Sub
```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, ResultRow As Long, SelRow As Long, PayRow
  As Long, LastItemRow As Long
3 Dim PayrollRow As Long, PaidEmpRow As Long, PaidItemRow As Long, PayEmpRow As Long,
  ItemDBRow As Long
4 Dim EmplRow As Long
5
6 Sub PayEmpl_Load()
7     With Payroll
8         .Range(
9             "B16,B19,N4,P4,N6,P6,N8,P8,M12:O999,S12:S999,W12:Y22,S26:U37,W26:Y37,AA12:AA37" )
10        .ClearContents
11        If .Range("B13" ).Value = Empty Then
12            MsgBox "Please select on an Employee to load"
13            Exit Sub
14        End If
15        .Range("N6" ).Value = .Range("I4" ).Value 'Payroll Date
16        PaidEmpRow = .Range("B13" ).Value
17        If .Range("B8" ).Value = Empty Then
18            MsgBox "Please select a correct employee to load"
19            Exit Sub
20        End If
21        Application.ScreenUpdating = False
22        EmplRow = .Range("B8" ).Value 'Employee DB Row
23        .Range("N4" ).Value = PaidEmpl.Range("F" & PaidEmpRow).Value 'Employee Name
24        .Range("P4" ).Value = PaidEmpl.Range("G" & PaidEmpRow).Value 'Pay Type
25        If PaidEmpl.Range("H" & PaidEmpRow).Value = Empty Then 'Get Employee Defaults
26            If .Range("P4" ).Value = "Hourly" Then
27                .Range("N8" ).Value = EmployeeDB.Range("E" & EmplRow).Value 'Regular
28                Hourly Pay
29            Else 'Salary (based on pay Frequency)
30                On Error Resume Next
31                .Range("N8" ).Value = EmployeeDB.Range("E" & EmplRow).Value / .Range(
32                    "B11" ).Value 'Pay Period Salary Amount
33                On Error GoTo 0
34            End If
35        Else 'Use Saved Values
36            .Range("N8" ).Value = PaidEmpl.Range("H" & PaidEmpRow).Value 'Saved
37            Amount/Rate
38        End If
39        If .Range("P4" ).Value = "Hourly" And PaidEmpl.Range("I" & PaidEmpRow).Value =
40            Empty Then 'Get Employee OT Rate
41            .Range("P8" ).Value = EmployeeDB.Range("F" & EmplRow).Value
42        Else 'Previously Saved
43            .Range("P8" ).Value = PaidEmpl.Range("I" & PaidEmpRow).Value 'SavedOT Rate
44        End If
45        .Range("P6" ).Value = PaidEmpl.Range("J" & PaidEmpRow).Value 'PTO Taken
46        .Range("U9" ).Copy
47        .Range("U12:U22" ).PasteSpecial xlPasteFormulas 'Re-Add in Tax Calculation
48        Formulas
49        Employee_AddUpdatePicture 'Show Picture (if any)
50        PayEmpl_LoadTimesheetEntries 'Load Timesheet Entries
51        PayEmpl_LoadPayItems 'Load Payroll Items
52        Application.ScreenUpdating = True
53    End With
54 End Sub
55
56 Sub PayEmpl_LoadPayItems()
57     With PayItems
58         LastRow = .Range("A99999" ).End(xlUp).Row
59         If LastRow < 4 Then Exit Sub
60         .Range("O3" ).Value = "Tax" 'Set Tax Criteria
61     End With
62 End Sub

```

```

1 2
53 .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "M2:O3" ), CopyToRange:=.Range("Q2:U2" ), Unique:=True
54 LastResultRow = .Range("Q99999" ).End(xlUp).Row
55 If LastResultRow < 3 Then GoTo NoTaxItems
56 For ResultRow = 3 To LastResultRow
57     PayRow = .Range("U" & ResultRow)
58     Payroll.Range("T" & PayRow).Value = .Range("R" & ResultRow).Value 'Tax Item
        Name
59     Payroll.Range("U" & PayRow).Value = .Range("S" & ResultRow).Value 'Tax
        Deduction Amount
60     Payroll.Range("W" & PayRow).Value = .Range("T" & ResultRow).Value 'Item DB
        Row
61 Next ResultRow
62 NoTaxItems:
63
64 'Add Other Payroll Deductions
65 .Range("O3" ).Value = "Deduction" 'Set Deduction Criteria
66 .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "N2:O3" ), CopyToRange:=.Range("X2:X2" ), Unique:=True
67 .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "M2:O3" ), CopyToRange:=.Range("Q2:U2" ), Unique:=True
68 LastResultRow = .Range("X99999" ).End(xlUp).Row
69 If LastResultRow < 3 Then GoTo NoDeductions
70 Payroll.Range("X12:X" & LastResultRow + 9).Value = .Range("X3:X" &
    LastResultRow).Value 'Other Deduction Name
71 Payroll.Range("Y12:Y" & LastResultRow + 9).Value = .Range("Y3:Y" &
    LastResultRow).Value 'Amount
72 Payroll.Range("AA12:AA" & LastResultRow + 9).Value = .Range("ZX3:Z" &
    LastResultRow).Value 'DB Row
73 NoDeductions:
74
75 'Add Other Payroll Additions
76 .Range("O3" ).Value = "Addition" 'Set Addition Criteria
77 .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "N2:O3" ), CopyToRange:=.Range("X2:X2" ), Unique:=True
78 .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "M2:O3" ), CopyToRange:=.Range("Q2:U2" ), Unique:=True
79 LastResultRow = .Range("X99999" ).End(xlUp).Row
80 If LastResultRow < 3 Then GoTo NoAdditions
81 Payroll.Range("X26:X" & LastResultRow + 23).Value = .Range("X3:X" &
    LastResultRow).Value 'Other Addition Name
82 Payroll.Range("Y26:Y" & LastResultRow + 23).Value = .Range("Y3:Y" &
    LastResultRow).Value 'Amount
83 Payroll.Range("AA26:AA" & LastResultRow + 23).Value = .Range("ZX3:Z" &
    LastResultRow).Value 'DB Row
84 NoAdditions:
85
86 'Add PTO Items
87 .Range("O3" ).Value = "PTO" 'Set PTO Criteria
88 .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "N2:O3" ), CopyToRange:=.Range("Q2:U2" ), Unique:=True
89 LastResultRow = .Range("R99999" ).End(xlUp).Row
90 If LastResultRow < 3 Then GoTo NoPTOItems
91 If LastResultRow < 4 Then GoTo NoSort
92 With .Sort 'Sort PTO From Newest to Oldest
93     .SortFields.Clear
94     .SortFields.Add Key:=PayItems.Range("Q3" ), SortOn:=xlSortOnValues, Order:=
        xlDescending, DataOption:=xlSortNormal 'Sort
95     .SetRange PayItems.Range("Q3:U" & LastResultRow) 'Set Range
96     .Apply 'Apply Sort
97 End With

```

1 2


```

1 2
98 NoSort:
99 Payroll.Range("T26:T" & LastResultRow + 23).Value = .Range("R3:R" &
LastResultRow).Value 'PTO Dates
100 Payroll.Range("U26:U" & LastResultRow + 23).Value = .Range("S3:S" &
LastResultRow).Value 'PTO Day Qty
101 NoPTOItems:
102 If ActiveSheet.Name = "Payroll" Then Payroll.Range("N8" ).Select
103 End With
104 End Sub

105
106 Sub PayEmpl_LoadTimesheetEntries()
107 Payroll.Range("M12:O9999,S12:S9999" ).ClearContents 'Clear Existing Timesheet Data
108 With Timesheet
109 LastRow = .Range("A9999" ).End(xlUp).Row
110 If LastRow < 4 Then Exit Sub
111 .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"L2:N3" ), CopyToRange:=.Range("R2:U2" ), Unique:=True
112 LastResultRow = .Range("R99999" ).End(xlUp).Row
113 If LastResultRow < 3 Then Exit Sub
114 If LastRow < 4 Then GoTo SkipSort
115 With .Sort 'Sort By Entry Date
116 .SortFields.Clear
117 .SortFields.Add Key:=Timesheet.Range("R3" ), SortOn:=xlSortOnValues, Order:=
xlAscending, DataOption:=xlSortNormal 'Sort
118 .SetRange Timesheet.Range("R3:U" & LastResultRow) 'Set Range
119 .Apply 'Apply Sort
120 End With
121 SkipSort:
122 Payroll.Range("M12:O" & LastResultRow + 9).Value = .Range("R3:T" &
LastResultRow).Value 'Add Entry Date, From & To
123 Payroll.Range("S12:S" & LastResultRow + 9).Value = .Range("U3:U" &
LastResultRow).Value 'Add DB Row
124 End With
125 End Sub

126
127 Sub PayEmpl_Previous()
128 With Payroll
129 If .Range("B10" ).Value = Empty Then Exit Sub
130 SelRow = .Range("B10" ).Value 'Selected Row
131 If SelRow = 13 Then
132 MsgBox "You are on the first employee"
133 Exit Sub
134 Else
135 .Range("H" & SelRow - 1).Select
136 End If
137 End With
138 End Sub

139
140 Sub PayEmpl_SaveNext()
141 With Payroll
142 If .Range("B3" ).Value = Empty Or .Range("B8" ).Value = Empty Then
143 MsgBox "Please make sure to select or add a payroll and also select or add an
Employee"
144 Exit Sub
145 End If
146 SelRow = .Range("B10" ).Value 'Selected Employee Row
147 If .Range("B13" ).Value = Empty Then 'New Paid Employee DB Row
148 PaidEmpRow = PaidEmpl.Range("A99999" ).End(xlUp).Row + 1
149 .Range("B13" ).Value = PaidEmpRow 'Set DB Row
150 PaidEmpl.Range("A" & PaidEmpRow).Value = .Range("B14" ).Value 'Set Next Paid
ID

```

1 2 3

```

151     1   2   3   PaidEmpl.Range("Q" & PaidEmpRow).Value = "=Row()" 'Set DB Row
152     } Else
153     PaidEmpRow = .Range("B13").Value 'Existing
154     .Range("U9").Copy 'Copy Tax Calc Formula
155     .Range("U12:U22").PasteSpecial xlPasteFormulas 'Re-Add in Tax Calculation
156     } End If
157     .Range("G" & SelRow).Value = Chr(254) 'Mark as paid
158     .Range("J" & SelRow).Value = .Range("V8").Value 'Set Gross Pay
159     .Range("K" & SelRow).Value = .Range("Z8").Value 'Set Net Pay
160     PaidEmpl.Range("B" & PaidEmpRow).Value = .Range("B2").Value 'Payroll ID
161     PaidEmpl.Range("C" & PaidEmpRow).Value = .Range("N6").Value 'Payroll Date
162     PaidEmpl.Range("D" & PaidEmpRow).Value = .Range("B7").Value 'Employee ID
163     PaidEmpl.Range("E" & PaidEmpRow).Value = Chr(254) 'Completed Status
164     PaidEmpl.Range("F" & PaidEmpRow).Value = .Range("N4").Value 'Employee Name
165     PaidEmpl.Range("G" & PaidEmpRow).Value = .Range("P4").Value 'Pay Type
166     PaidEmpl.Range("H" & PaidEmpRow).Value = .Range("N8").Value 'Amount/Rate
167     PaidEmpl.Range("I" & PaidEmpRow).Value = .Range("P8").Value 'OT Rate
168     PaidEmpl.Range("J" & PaidEmpRow).Value = .Range("P6").Value 'PTO Taken
169     PaidEmpl.Range("K" & PaidEmpRow).Value = .Range("V4").Value 'Regular Hours
170     PaidEmpl.Range("L" & PaidEmpRow).Value = .Range("V5").Value 'OT Hours
171     PaidEmpl.Range("M" & PaidEmpRow).Value = .Range("V6").Value 'Reg. Pay
172     PaidEmpl.Range("N" & PaidEmpRow).Value = .Range("V7").Value 'OT Pay
173     PaidEmpl.Range("O" & PaidEmpRow).Value = .Range("V8").Value 'Gross Pay
174     PaidEmpl.Range("P" & PaidEmpRow).Value = .Range("Z8").Value 'Net Pay
175
176     '''Save Pay Items
177
178     'Save/Update Tax Items
179     LastItemRow = .Range("T23").End(xlUp).Row
180     If LastItemRow < 12 Then GoTo NoTaxItems
181     For PaidItemRow = 12 To LastItemRow
182     If .Range("T" & PaidItemRow).Value = 0 Then Exit For
183     If .Range("W" & PaidItemRow).Value = "" Then
184     ItemDBRow = PayItems.Range("A99999").End(xlUp).Row + 1 'First Avail Row
185     .Range("W" & PaidItemRow).Value = ItemDBRow 'Set DB Row
186     PayItems.Range("A" & ItemDBRow).Value = .Range("B2").Value 'Payroll ID
187     PayItems.Range("E" & ItemDBRow).Value = "Tax" 'Set Item Type
188     PayItems.Range("H" & ItemDBRow).Value = PaidItemRow
189     PayItems.Range("I" & ItemDBRow).Value = "=Row()"
190     } Else 'Previously Saved
191     ItemDBRow = .Range("W" & PaidItemRow).Value 'Existing DB Row
192     } End If
193     PayItems.Range("B" & ItemDBRow).Value = .Range("N6").Value 'Pay Date
194     PayItems.Range("C" & ItemDBRow).Value = .Range("B7").Value 'Empl. ID
195     PayItems.Range("D" & ItemDBRow).Value = .Range("N4").Value 'Empl. Name
196     PayItems.Range("F" & ItemDBRow).Value = .Range("T" & PaidItemRow).Value
197     'Tax Item Name
198     PayItems.Range("G" & ItemDBRow).Value = .Range("U" & PaidItemRow).Value
199     'Tax Item Amount
200     Next PaidItemRow
201     NoTaxItems:
202
203     'Save/Update Deductions
204     LastItemRow = .Range("X23").End(xlUp).Row
205     If LastItemRow < 12 Then GoTo NoDeductions
206     For PaidItemRow = 12 To LastItemRow
207     If .Range("AA" & PaidItemRow).Value = "" Then
208     ItemDBRow = PayItems.Range("A99999").End(xlUp).Row + 1 'First Avail Row
209     .Range("AA" & PaidItemRow).Value = ItemDBRow 'Set DB Row
210     PayItems.Range("A" & ItemDBRow).Value = .Range("B2").Value 'Payroll ID

```

```

1 2 3 4
209 PayItems.Range("E" & ItemDBRow).Value = "Deduction" 'Set Item Type
210 PayItems.Range("H" & ItemDBRow).Value = PaidItemRow
211 PayItems.Range("I" & ItemDBRow).Value = "=Row()"
212 Else 'Previously Saved
213     ItemDBRow = .Range("AA" & PaidItemRow).Value 'Existing DB Row
214 End If
215 PayItems.Range("B" & ItemDBRow).Value = .Range("N6" ).Value 'Pay Date
216 PayItems.Range("C" & ItemDBRow).Value = .Range("B7" ).Value 'Empl. ID
217 PayItems.Range("D" & ItemDBRow).Value = .Range("N4" ).Value 'Empl. Name
218 PayItems.Range("F" & ItemDBRow).Value = .Range("X" & PaidItemRow).Value
    'Deduction Item Name
219 PayItems.Range("G" & ItemDBRow).Value = .Range("Y" & PaidItemRow).Value
    'Deduction Amount
220 Next PaidItemRow
221 NoDeductions:
222
223 'Save/Update Additions
224 LastItemRow = .Range("X38" ).End(xlUp).Row
225 If LastItemRow < 26 Then GoTo NoAdditions
226 For PaidItemRow = 26 To LastItemRow
227     If .Range("AA" & PaidItemRow).Value = "" Then
228         ItemDBRow = PayItems.Range("A99999" ).End(xlUp).Row + 1 'First Avail Row
229         .Range("AA" & PaidItemRow).Value = ItemDBRow 'Set DB Row
230         PayItems.Range("A" & ItemDBRow).Value = .Range("B2" ).Value 'Payroll ID
231         PayItems.Range("E" & ItemDBRow).Value = "Addition" 'Set Item Type
232         PayItems.Range("H" & ItemDBRow).Value = PaidItemRow
233         PayItems.Range("I" & ItemDBRow).Value = "=Row()"
234     Else 'Previously Saved
235         ItemDBRow = .Range("AA" & PaidItemRow).Value 'Existing DB Row
236     End If
237     PayItems.Range("B" & ItemDBRow).Value = .Range("N6" ).Value 'Pay Date
238     PayItems.Range("C" & ItemDBRow).Value = .Range("B7" ).Value 'Empl. ID
239     PayItems.Range("D" & ItemDBRow).Value = .Range("N4" ).Value 'Empl. Name
240     PayItems.Range("F" & ItemDBRow).Value = .Range("X" & PaidItemRow).Value
        'Addition Item Name
241     PayItems.Range("G" & ItemDBRow).Value = .Range("Y" & PaidItemRow).Value
        'Addition Amount
242 Next PaidItemRow
243 NoAdditions:
244 'Save/Update PTO
245 If .Range("P6" ).Value <> Empty Then 'PTO Taken
246     If .Range("B15" ).Value = 0 Then 'New Pay Item Row
247         ItemDBRow = PayItems.Range("A99999" ).End(xlUp).Row + 1 'First Avail Row
248         PayItems.Range("A" & ItemDBRow).Value = .Range("B2" ).Value 'Payroll ID
249         PayItems.Range("E" & ItemDBRow).Value = "PTO" 'Set Item Type
250         PayItems.Range("I" & ItemDBRow).Value = "=Row()"
251     Else
252         ItemDBRow = .Range("B15" ).Value
253     End If
254     PayItems.Range("B" & ItemDBRow).Value = .Range("N6" ).Value 'Pay Date
255     PayItems.Range("C" & ItemDBRow).Value = .Range("B7" ).Value 'Empl. ID
256     PayItems.Range("D" & ItemDBRow).Value = .Range("N4" ).Value 'Empl. Name
257     PayItems.Range("F" & ItemDBRow).Value = .Range("N6" ).Value 'Pay Date (as
        Item Name)
258     PayItems.Range("G" & ItemDBRow).Value = .Range("P6" ).Value 'PTO Qty
259 End If
260
261
262 'Load Next Employee (If available)
263 If .Range("H" & SelRow + 1).Value = "" Then
264     MsgBox "You are at the final Employee"

```

1 2 3

```
265 | 1 2 3 Exit Sub
266 | | | End If
267 | | | .Range("H" & SelRow + 1).Select 'Select to load next employee
268 | | | End With
269 | | | End Sub
```

```

1 Option Explicit
2 Dim LastRow As Long, LastResultRow As Long, ResultRow As Long, SelRow As Long
3 Dim PayrollRow As Long, PaidEmpRow As Long, PaidItemRow As Long, PayEmpRow As Long
4 Dim PaidEmpID As Long
5 Dim PayFreq As String
6 Dim FromDt As Date, ToDt As Date
7
8 Sub Payroll_AddAllEmployees()
9     Payroll.Range("F13:L9999").ClearContents
10    With EmployeeDB
11        LastRow = .Range("A9999").End(xlUp).Row
12        If LastRow < 4 Then Exit Sub
13        .Range("A3:P" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
14            "R2:T3"), CopyToRange:=.Range("V2:Y2"), Unique:=True
15        LastResultRow = .Range("V9999").End(xlUp).Row
16        If LastResultRow < 3 Then Exit Sub
17        .Range("W3:W" & LastResultRow).Value = Chr(111) 'Set Empty Checkbox
18        If LastResultRow < 4 Then GoTo NoSort 'No Sort on Single Employee
19        With .Sort
20            .SortFields.Clear
21            .SortFields.Add Key:=Employees.Range("X3"), SortOn:=xlSortOnValues, Order:=
22                xlAscending, DataOption:=xlSortNormal 'Sort
23            .SetRange Employees.Range("V3:Y" & LastResultRow) 'Set Range
24            .Apply 'Apply Sort
25        End With
26        NoSort:
27        Payroll.Range("F13:I" & LastResultRow + 10).Value = .Range("V3:Y" &
28            LastResultRow).Value
29    End With
30 End Sub
31
32 Sub Payroll_AddNew()
33     With Payroll
34         .Range(
35             "B2,B5,B7,B10,B13,B16,B19,F13:L9999,N4,P4,N6,P6,N8,P8,M12:O999,S12:S999,W12:Y22,T2
36             6:U37,W26:Y37,AA12:AA37").ClearContents
37         .Range("I4").Value = Date 'Set Current Date Default Payroll Date
38         .Range("K4").Value = Admin.Range("D10").Value 'Set to first Payroll Status
39         .Range("I6").Value = .Range("B6").Value 'Next Payroll From Date
40         .Range("K6").Value = .Range("I6").Value + (365 / .Range("B11").Value) - 1
41         'Set Pay To Date
42         .Range("U9").Copy 'Copy Tax Formula
43         .Range("U12:U22").PasteSpecial xlPasteFormulas 'Paste Tax Formula
44         .Shapes("EmplPic").Visible = msoFalse
45         .Range("I11,K11").Value = "[All Types]" 'Clear Employee Filters
46     End With
47 End Sub
48
49 Sub Payroll_LoadList()
50     Payroll.Range("B5,C6:E9999").ClearContents 'Clear Existing PayrollDB
51     With PayrollDB
52         LastRow = .Range("A9999").End(xlUp).Row
53         If LastRow < 4 Then Exit Sub
54         .Range("A3:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
55             "J2:K3"), CopyToRange:=.Range("N2:P2"), Unique:=True
56         LastResultRow = .Range("N9999").End(xlUp).Row
57         If LastResultRow < 3 Then Exit Sub
58         If LastResultRow < 4 Then GoTo NoSort
59         'Sort Alphabetally
60         With .Sort
61             .SortFields.Clear

```

1 2 3

```

1 2 3
55     .SortFields.Add Key:=PayrollDB.Range("O3" ), SortOn:=xlSortOnValues, Order:=
        xlDescending, DataOption:=xlSortNormal 'Sort
56     .SetRange PayrollDB.Range("N3:P" & LastResultRow) 'Set Range
57     .Apply 'Apply Sort
58     End With
59     NoSort:
60     Payroll.Range("C6:E" & LastResultRow + 3).Value = .Range("N3:P" & LastResultRow
        ).Value 'Bring Over Payroll List
61     End With
62     With Payroll
63         On Error Resume Next
64         SelRow = .Range("C6:C9999" ).Find(.Range("B2" ).Value, , xlFormulas, xlWhole).Row
65         If SelRow <> 0 And .Range("B2" ).Value <> Empty Then .Range("B5" ).Value = SelRow
            'Set Selected Project Row
66         On Error GoTo 0
67     End With
68 End Sub
69 Sub Payroll_LoadPayroll()
70     With Payroll
71         .Range(
            "B7,B10,B13,B16,B19,F13:L9999,N4,P4,N6,P6,N8,P8,M12:O999,S12:S999,W12:Y22,S26:U37,
            W26:Y37,AA12:AA37" ).ClearContents
72         If .Range("B3" ).Value = Empty Then
73             MsgBox "Please select a correct payroll to load details"
74             Exit Sub
75         End If
76         PayrollRow = .Range("B3" ).Value
77         'Load Payroll Details
78         .Range("I4" ).Value = PayrollDB.Range("B" & PayrollRow).Value 'Pay Date
79         .Range("K4" ).Value = PayrollDB.Range("C" & PayrollRow).Value 'Status
80         .Range("I6" ).Value = PayrollDB.Range("D" & PayrollRow).Value 'From Date
81         .Range("K6" ).Value = PayrollDB.Range("E" & PayrollRow).Value 'To Date
82         With PaidEmpl
83             LastRow = .Range("A99999" ).End(xlUp).Row
84             If LastRow < 4 Then 'No Paid Employees, Add Employees From Employee List
85                 Payroll_AddAllEmployees
86                 Exit Sub
87             End If
88             .Range("A3:Q" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
                "V2:V3" ), CopyToRange:=.Range("AA2:AG2" ), Unique:=True
89             LastResultRow = .Range("AA99999" ).End(xlUp).Row
90             If LastResultRow < 3 Then 'No Paid Employees, Add Employees From Employee
                List
91                 Payroll_AddAllEmployees
92                 Exit Sub
93             End If
94
95             If LastResultRow < 4 Then GoTo NoSort
96             With .Sort
97                 .SortFields.Clear
98                 .SortFields.Add Key:=PaidEmpl.Range("AC3" ), SortOn:=xlSortOnValues, Order:=
                    xlAscending, DataOption:=xlSortNormal 'Sort
99                 .SetRange PaidEmpl.Range("AA3:AG" & LastResultRow) 'Set Range
100                .Apply 'Apply Sort
101            End With
102            NoSort:
103            Payroll.Range("F13:L" & LastResultRow + 10).Value = .Range("AA3:AG" &
                LastResultRow).Value 'Bring Over Payroll Employees
104        End With
105    End With

```

1

```

1
106 End Sub
107 Sub Payroll_SaveUpdate()
108     With Payroll
109         If .Range("I4").Value = Empty Or .Range("K4").Value = Empty Or .Range("I6").
110             Value = Empty Or .Range("K6").Value = Empty Then
111             MsgBox "Please make sure to add a Payroll Date, Status as well as From and To
112                 Dates before saving"
113             Exit Sub
114         End If
115         If .Range("B3").Value = Empty Then
116             PayrollRow = PayrollDB.Range("A9999").End(xlUp).Row + 1
117             .Range("B2").Value = .Range("B4").Value 'Next Payroll ID
118             PayrollDB.Range("A" & PayrollRow).Value = .Range("B2").Value 'Next Payroll
119             ID
120         Else 'Existing Payroll
121             PayrollRow = .Range("B3").Value 'Existing Row
122         End If
123         PayrollDB.Range("B" & PayrollRow).Value = .Range("I4").Value 'Pay Date
124         PayrollDB.Range("C" & PayrollRow).Value = .Range("K4").Value 'Status
125         PayrollDB.Range("D" & PayrollRow).Value = .Range("I6").Value 'From Date
126         PayrollDB.Range("E" & PayrollRow).Value = .Range("K6").Value 'To Date
127         PayrollDB.Range("F" & PayrollRow).Value = .Range("I8").Value 'Gross Amount
128         PayrollDB.Range("G" & PayrollRow).Value = .Range("K8").Value 'Net Amount
129
130         'Save/Update Paid Employees
131         LastRow = .Range("H9999").End(xlUp).Row 'Last Employee Row
132         If LastRow < 13 Then Exit Sub 'No Employees
133         PaidEmpID = .Range("B14").Value
134         For PayEmpRow = 13 To LastRow
135             If .Range("L" & PayEmpRow).Value = "" Then 'New Paid Employee
136                 PaidEmpRow = PaidEmpl.Range("A9999").End(xlUp).Row + 1
137                 PaidEmpl.Range("A" & PaidEmpRow).Value = PaidEmpID 'Set Next Paid
138                 employee ID
139                 PaidEmpl.Range("Q" & PaidEmpRow).Value = "=Row()" 'Set DB Row
140                 PaidEmpID = PaidEmpID + 1 'Increment ID
141                 .Range("L" & PayEmpRow).Value = PaidEmpRow 'Set Paid Emp. DB Row
142             Else 'Existing Paid Employee
143                 PaidEmpRow = .Range("L" & PayEmpRow).Value 'Set Existing Row
144             End If
145             PaidEmpl.Range("B" & PaidEmpRow).Value = .Range("B2").Value 'Payroll ID
146             PaidEmpl.Range("C" & PaidEmpRow).Value = .Range("I4").Value 'Pay Date
147             PaidEmpl.Range("D" & PaidEmpRow).Value = .Range("F" & PayEmpRow).Value
148             'Employee ID
149             PaidEmpl.Range("E" & PaidEmpRow).Value = .Range("G" & PayEmpRow).Value
150             'Approved Status
151             PaidEmpl.Range("F" & PaidEmpRow).Value = .Range("H" & PayEmpRow).Value
152             'Empl. Name
153             PaidEmpl.Range("G" & PaidEmpRow).Value = .Range("I" & PayEmpRow).Value 'Pay
154             Type
155         Next PayEmpRow
156         MsgBox "Payroll Saved"
157     End With
158     Payroll_LoadList 'Reload Payroll List
159 End Sub
160
161 Sub PrintPayStub()
162     Payroll.PrintOut , , , False, True, , , , False
163 End Sub
164
165
166

```

```

1 Option Explicit
2 Dim ProdRow As Long, ProdCol As Long, LastRow As Long, LastResultRow As Long
3 Dim ProdFld As control
4
5 Sub Product_AddNew()
6     Products.Range("B3,B4" ).ClearContents 'Clear Selected Product Row and Prod DB Row
7     ProductFrm.Show
8 End Sub
9
10 Sub Product_AddPicture()
11     Dim ProdPic As FileDialog
12     Dim PicName As String, PicPath As String, PicFolder As String
13
14     If [ProductFolder] = "" Then
15         MsgBox "Please select a folder for product pictures"
16         BrowseForProductPicFolder
17         If [ProductFolder] = "" Then Exit Sub
18     End If
19     PicFolder = [ProductFolder] 'Set Picture Folder
20     Set ProdPic = Application.FileDialog(msoFileDialogFilePicker)
21     With ProdPic
22         .Title = "Select Product Picture"
23         .Filters.Add "Select Picture", "*.jpg, *.png, *.jpeg, *.gif", 1
24         If .Show <> -1 Then GoTo NoSelection
25         PicPath = .SelectedItems(1) 'Entire Picture Path
26     End With
27     PicName = Dir(PicPath) 'Set Picture Name
28     ProductFrm.Field4.Value = PicName 'Set Picture Name
29     'Copy Product Picture to picture folder
30     On Error Resume Next
31     FileCopy PicPath, PicFolder & "\" & PicName
32     On Error GoTo 0
33     NoSelection:
34 End Sub
35 Sub Product_ClearFilter()
36     Products.Range("F2:H2" ).ClearContents
37 End Sub
38
39 Sub Product_Delete()
40     If MsgBox("Are you sure you want to delete this product?" , vbYesNo, "Delete
Product" ) = vbNo Then Exit Sub
41     If Products.Range("B4" ).Value = Empty Then Exit Sub
42     ProdRow = Products.Range("B4" ).Value 'Product Row
43     For ProdCol = 2 To 10
44         ProductDB.Cells(ProdRow, ProdCol).ClearContents
45     Next ProdCol
46     Products.Range("B3,B4" ).ClearContents
47     Product_RefreshList
48 End Sub
49
50
51 Sub Product_Edit()
52     With Products
53         If .Range("B4" ).Value = Empty Then
54             MsgBox "Please select on a product to edit"
55             Exit Sub
56         End If
57         ProdRow = .Range("B4" ).Value 'Product Row
58         With ProductFrm
59             For ProdCol = 2 To 7
60                 Set ProdFld = .Controls("Field" & ProdCol - 1)

```

1 2 3 4


```

1 2 3 4
61 ProdFld.Value = ProductDB.Cells(ProdRow, ProdCol).Value 'Add Product Data
62 Next ProdCol
63 .Show 'Display User Form
64 End With
65 End With
66 End Sub
67 Sub Product_RefreshList()
68 With ProductDB
69 Products.Range("D4:M9999").ClearContents
70 LastRow = .Range("A9999").End(xlUp).Row
71 If LastRow < 4 Then Exit Sub
72 Application.ScreenUpdating = False
73 .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
74 "M2:N3" ), CopyToRange:=.Range("P2:Y2" ), Unique:=True
75 LastResultRow = .Range("P9999").End(xlUp).Row
76 If LastResultRow < 3 Then
77 Application.ScreenUpdating = True
78 Exit Sub
79 End If
80 If LastResultRow < 4 Then GoTo NoSort 'No Sort on a single item
81 'Sort By ID
82 With .Sort
83 .SortFields.Clear
84 .SortFields.Add Key:=ProductDB.Range("P3" ), SortOn:=xlSortOnValues, Order:=
85 xlAscending, DataOption:=xlSortNormal 'Sort
86 .SetRange ProductDB.Range("P3:Y" & LastResultRow) 'Set Range
87 .Apply 'Apply Sort
88 End With
89 NoSort:
90 Products.Range("D4:M" & LastResultRow + 1).Value = .Range("P3:Y" &
91 LastResultRow).Value
92 Products.Range("F2").Select
93 Application.ScreenUpdating = True
94 End With
95 End Sub
96 Sub Product_SaveUpdate()
97 'Check For Required field
98 With ProductFrm
99 If .Field1.Value = Empty Then
100 MsgBox "Please make sure to add a Product Name before saving"
101 Exit Sub
102 End If
103 If Products.Range("B4").Value = Empty Then 'New Product
104 ProdRow = ProductDB.Range("A9999").End(xlUp).Row + 1 'First Avail. Row
105 ProductDB.Range("A" & ProdRow).Value = Products.Range("B5").Value 'Next
106 Product ID
107 ProductDB.Range("H" & ProdRow).Value =
108 "=SUMIFS(OrdItem_Qty,OrdItem_ProdNm,C" & ProdRow & ",OrdItem_Type," & Chr(34)
109 ) & "PURCHASE" & Chr(34) & ")-SUMIFS(OrdItem_Qty,OrdItem_ProdNm,C" & ProdRow
110 & ",OrdItem_Type," & Chr(34) & "INVOICE" & Chr(34) & ")" 'Qty On Hand
111 ProductDB.Range("I" & ProdRow).Value = "=F4*H4" 'Value On Hand
112 ProductDB.Range("J" & ProdRow).Value = "=Row()"
113 Else 'Existing Prod.
114 ProdRow = Products.Range("B4").Value 'Product Row
115 End If
116 For ProdCol = 2 To 7
117 Set ProdFld = .Controls("Field" & ProdCol - 1)
118 ProductDB.Cells(ProdRow, ProdCol).Value = ProdFld.Value
119 Next ProdCol
120 End With
121 Unload ProductFrm

```

1

```
1
115 'Update List Of Product Names (Sorted)
116 With ProductDB
117     LastRow = .Range("A99999" ).End(xlUp).Row
118     If LastRow < 4 Then Exit Sub
119     On Error Resume Next
120     .Names("Criteria" ).Delete
121     On Error GoTo 0
122     .Range("C3:C" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
123         "L2:L2" ), Unique:=True
124     With .Sort
125         .SortFields.Clear
126         .SortFields.Add Key:=ProductDB.Range("L3" ), SortOn:=xlSortOnValues, Order:=
127             xlAscending, DataOption:=xlSortNormal 'Sort
128     End With
129     .SetRange ProductDB.Range("L3:L" & LastRow) 'Set Range
130     .Apply 'Apply Sort
131 End With
132 Product_RefreshList
133 End Sub
```

```

1 Option Explicit
2 Dim ResRow As Long, SelRow As Long
3
4 Sub Resource_AddNew()
5     With CapPlan
6         .Range("B2" ).ClearContents 'Sel Selected Resource ID
7         ResourceFrm.ActiveStatus.Value = True 'Active Status set to true/checked
8         ResourceFrm.Show
9     End With
10 End Sub
11
12 Sub Resource_AddPicture()
13     Dim ResPic As FileDialog
14     Dim PicName As String, PicPath As String, PicFolder As String
15
16     If [PictureFolder] = "" Then
17         MsgBox "Please select a folder to place your Resource Pictures"
18         BrowseForPicFolder
19         If [PictureFolder] = "" Then Exit Sub
20     End If
21     PicFolder = [PictureFolder] 'Set Picture Folder
22     Set ResPic = Application.FileDialog(msoFileDialogFilePicker)
23     With ResPic
24         .Title = "Select Resource Picture"
25         .Filters.Add "Select Picture", "*.jpg, *.png, *.jpeg, *.giff", 1
26         If .Show <> -1 Then GoTo NoSelection
27         PicPath = .SelectedItems(1) 'Entire Picture Path
28     End With
29     PicName = Dir(PicPath) 'Set Picture Name
30     ResourceFrm.Picture.Value = PicName 'Set Picture Name
31     'Copy Resource Picture to picture folder
32     On Error Resume Next
33     FileCopy PicPath, PicFolder & "\" & PicName
34     On Error GoTo 0
35     NoSelection:
36 End Sub
37
38 Sub Resource_Edit()
39     Dim ResType As String, TypeRow As Long
40     With CapPlan
41         If .Range("B5" ).Value = Empty Then Exit Sub
42         SelRow = .Range("B5" ).Value 'Selected Row
43         If .Range("C" & SelRow).Value = Empty Then Exit Sub 'Exit on no resource ID
44         .Range("B2" ).Value = .Range("C" & SelRow).Value 'Set Resouce ID
45         If .Range("B3" ).Value = Empty Then 'No DB Row, Add New Resource
46             Resource_AddNew
47         Else 'Load resources
48             ResRow = .Range("B3" ).Value
49             With ResourceFrm
50                 .ResName.Value = Resources.Range("B" & ResRow).Value 'Resource Name
51                 .ResType.Value = Resources.Range("C" & ResRow).Value 'Resource Type
52                 On Error Resume Next
53                 TypeRow = Admin.Range("ResourceTypes" ).Find(.ResType.Value, , xlValues,
54                     xlWhole).Row
55                 On Error GoTo 0
56                 If TypeRow <> 0 Then .LblDefCap.Caption = Admin.Range("C" & TypeRow).Value
57                     & " Capacity" Else .LblDefCap.Caption = "Default Capacity"
58                 .DefCap.Value = Resources.Range("E" & ResRow).Value 'Default Capacity
59                 .CostUnit.Value = Resources.Range("F" & ResRow).Value 'Cost Per Unit
60                 .SalesUnit.Value = Resources.Range("G" & ResRow).Value 'Sales Per Unit
61                 .ActiveStatus.Value = Resources.Range("H" & ResRow).Value 'Active Status

```

1 2 3 4

```

1 2 3 4
60 .Picture.Value = Resources.Range("I" & ResRow).Value 'Picture
61 .Show 'Display User Form
62 End With
63 End If
64 End With
65
66 End Sub
67 Sub Resource_SaveUpdate()
68 'Check For Required field
69 Dim ResType As String, TypeRow As Long
70 With ResourceFrm
71 If .ResName.Value = "" Or .ResType.Value = Empty Then
72 MsgBox "Please make sure to have both a Resource Name and Resource Type, as
they are required"
73 Exit Sub
74 End If
75 If CapPlan.Range("B3" ).Value = Empty Then 'New Resource
76 ResRow = Resources.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
77 CapPlan.Range("B2" ).Value = CapPlan.Range("B4" ).Value 'Next Resource ID
78 Resources.Range("A" & ResRow).Value = CapPlan.Range("B2" ).Value
79 Else 'Existing Res.
80 ResRow = CapPlan.Range("B3" ).Value 'Resource Row
81 End If
82 Resources.Range("B" & ResRow).Value = .ResName.Value 'Resource Name
83 Resources.Range("C" & ResRow).Value = .ResType.Value 'Resource Type
84 ResType = .ResType.Value 'Resource Type
85 On Error Resume Next
86 TypeRow = Admin.Range("ResourceTypes" ).Find(ResType, , xlValues, xlWhole).Row
'Resource Type Row
87 On Error GoTo 0
88 If TypeRow <> 0 Then Resources.Range("D" & ResRow).Value = Admin.Range("C" &
TypeRow).Value 'Unit Of Measure
89 Resources.Range("E" & ResRow).Value = .DefCap.Value 'Default Capacity
90 Resources.Range("F" & ResRow).Value = .CostUnit.Value 'Cost Per Unit
91 Resources.Range("G" & ResRow).Value = .SalesUnit.Value 'Sales Per Unit
92 Resources.Range("H" & ResRow).Value = .ActiveStatus.Value 'Active Status
93 Resources.Range("I" & ResRow).Value = .Picture.Value 'Picture
94 End With
95 ResourceFrm.Hide 'Hide Form
96 Unload ResourceFrm
97 CapSched_Refresh 'Refresh Capacity Scheduler
98 End Sub

```

```
1 Option Explicit
2
3 Sub Toolbar_Admin(control As IRibbonControl)
4     Admin.Activate
5 End Sub
6
7 Sub Toolbar_Customers(control As IRibbonControl)
8     Customers.Activate
9 End Sub
10
11
12 Sub Toolbar_Dashboard(control As IRibbonControl)
13     Dashboard.Activate
14 End Sub
15
16 Sub Toolbar_Employees(control As IRibbonControl)
17     Employees.Activate
18 End Sub
19
20
21 Sub Toolbar_Invoice(control As IRibbonControl)
22     Orders.Activate
23     OrderTab_Invoice
24 End Sub
25
26 Sub Toolbar_Payroll(control As IRibbonControl)
27     Payroll.Activate
28 End Sub
29
30 Sub Toolbar_Products(control As IRibbonControl)
31     Products.Activate
32 End Sub
33
34
35 Sub Toolbar_PurchaseOrders(control As IRibbonControl)
36     Orders.Activate
37     OrderTab_PO
38 End Sub
39
40 Sub Toolbar_Transactions(control As IRibbonControl)
41     Transactions.Activate
42 End Sub
43
44 Sub Toolbar_Users(control As IRibbonControl)
45     Users.Activate
46 End Sub
47
48 Sub Toolbar_Vendors(control As IRibbonControl)
49     Vendors.Activate
50 End Sub
51
52 Sub Toolbar_WorkOrders(control As IRibbonControl)
53     Orders.Activate
54     OrderTab_WO
55 End Sub
```

```

1  Option Explicit
2
3  Sub GenerateTreeView()
4      Dim OrdRow As Long, LastRow As Long, LastResultRow As Long, CustRow As Long
5      Dim OrdType As String, CustName As String, OrdID As String
6      Dim OrderSht As Worksheet
7
8      Set OrderSht = ThisWorkbook.Sheets("Order List" )
9      With OrderSht
10         LastRow = .Range("A99999" ).End(xlUp).Row 'Last Row Of Data
11         If LastRow < 4 Then Exit Sub
12         OrdersTreeView.TreeView2.Nodes.Clear 'Clear all existing data (nodes) from tree
13         view
14         On Error Resume Next
15         .Names("Criteria" ).Delete
16         On Error GoTo 0
17         ' Application.ScreenUpdating = False
18         ' Application.Calculation = xlCalculationManual
19         .Range("C3:C" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
20         "W2" ), Unique:=True
21         LastResultRow = .Range("W99999" ).End(xlUp).Row
22         If LastResultRow < 3 Then GoTo NoData
23         For CustRow = 3 To LastResultRow
24             With OrdersTreeView.TreeView2.Nodes
25                 .Add Key:=OrderSht.Range("W" & CustRow).Value, Text:=OrderSht.Range("W" &
26                 CustRow).Value 'Add Customer
27                 If OrdersTreeView.IncWorkOrders.Value = True Then .Add Relative:=OrderSht.
28                 Range("W" & CustRow).Value, Relationship:=tvwChild, Key:=OrderSht.Range(
29                 "W" & CustRow).Value & "-Work Orders" , Text:="Work Orders" , Image:=
30                 "Work Orders"
31                 If OrdersTreeView.IncPurchases.Value = True Then .Add Relative:=OrderSht.
32                 Range("W" & CustRow).Value, Relationship:=tvwChild, Key:=OrderSht.Range(
33                 "W" & CustRow).Value & "-Purchases" , Text:="Purchases" , Image:=
34                 "Purchases"
35                 If OrdersTreeView.IncInvoices.Value = True Then .Add Relative:=OrderSht.
36                 Range("W" & CustRow).Value, Relationship:=tvwChild, Key:=OrderSht.Range(
37                 "W" & CustRow).Value & "-Invoices" , Text:="Invoices" , Image:="Invoices"
38             End With
39         Next CustRow
40
41         For OrdRow = 4 To LastRow
42             OrdType = Application.WorksheetFunction.Proper(.Range("B" & OrdRow).Value) &
43             "s" 'Order Type
44             CustName = .Range("C" & OrdRow).Value 'Customer Name
45             OrdID = .Range("A" & OrdRow).Value 'Order ID
46             With OrdersTreeView.TreeView2.Nodes
47                 If OrdersTreeView.IncWorkOrders.Value = False And OrdType = "Work Orders"
48                 Then GoTo NextOrder
49                 If OrdersTreeView.IncPurchases.Value = False And OrdType = "Purchases"
50                 Then GoTo NextOrder
51                 If OrdersTreeView.IncInvoices.Value = False And OrdType = "Invoices" Then
52                 GoTo NextOrder
53                 .Add Relative:=CustName & "-" & OrdType, Relationship:=tvwChild, Key:=
54                 "Order" & OrdID, Text:=Left(OrdType, Len(OrdType) - 1) & " #" & OrdID
55                 'Add Order
56             End With
57             NextOrder:
58         Next OrdRow
59     End With
60 End Sub

```

```
1 2
44 NoData:
45 Application.ScreenUpdating = True
46 Application.Calculation = xlCalculationAutomatic
47 End With
48 End Sub
49
50
51 Sub PrintSelectedOrders()
52 Dim OrdCount As Long
53 Dim OrdID As String
54 Dim Node As Node
55 With OrdersTreeView
56 If .CheckBoxes.Value = False Then
57 MsgBox "Please make sure to select the 'Enable Checkboxes', then select those
58 orders to be printed"
59 Exit Sub
60 End If
61 For Each Node In .TreeView2.Nodes
62 If InStr(Node.Text, "#" ) > 0 And Node.Checked = True Then
63 OrdCount = OrdCount + 1 'Increment the order count
64 OrdID = Replace(Node.Key, "Order" , "" ) 'Extract Order ID
65 Orders.Range("Q5" ).Value = OrdID 'Set Order ID/#
66 Order_Print
67 End If
68 Next Node
69 End With
70 MsgBox OrdCount & " orders have been printed"
71 End Sub
72
73 Sub ShowTrainTreeView()
74 OrdersTreeView.Show
75 End Sub
```

```

1 Option Explicit
2 Dim TransRow As Long, TransCol As Long, LastRow As Long, LastResultRow As Long
3 Dim TransFld As control
4
5 Sub Transaction_AddNew()
6     Transactions.Range("B3,B4" ).ClearContents 'Clear Selected Transaction Row and
7     Trans DB Row
8     TransactFrm.Show
9 End Sub
10 Sub Transaction_ClearFilter()
11     Transactions.Range("F2:G2" ).ClearContents
12 End Sub
13
14 Sub Transaction_Delete()
15     If MsgBox("Are you sure you want to delete this Transaction?" , vbYesNo, "Delete
16     Transaction" ) = vbNo Then Exit Sub
17     If Transactions.Range("B4" ).Value = Empty Then Exit Sub
18     TransRow = Transactions.Range("B4" ).Value 'Transaction Row
19     For TransCol = 2 To 8
20         TransDB.Cells(TransRow, TransCol).ClearContents
21     Next TransCol
22     Transactions.Range("B3,B4" ).ClearContents
23     Transaction_RefreshList
24 End Sub
25
26
27 Sub Transaction_Edit()
28     With Transactions
29         If .Range("B4" ).Value = Empty Then
30             MsgBox "Please select on a Transaction to edit"
31             Exit Sub
32         End If
33         TransRow = .Range("B4" ).Value 'Transaction Row
34         With TransactFrm
35             For TransCol = 2 To 7
36                 Set TransFld = .Controls("Field" & TransCol - 1)
37                 TransFld.Value = TransDB.Cells(TransRow, TransCol).Value 'Add Transaction
38                 Data
39             Next TransCol
40             .Show 'Display User Form
41         End With
42     End With
43 End Sub
44 Sub Transaction_RefreshList()
45     With TransDB
46         Transactions.Range("D4:K9999" ).ClearContents
47         LastRow = .Range("A99999" ).End(xlUp).Row
48         If LastRow < 4 Then Exit Sub
49         Application.ScreenUpdating = False
50         .Range("A3:H" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
51         "M2:N3" ), CopyToRange:=.Range("Q2:X2" ), Unique:=True
52         LastResultRow = .Range("Q99999" ).End(xlUp).Row
53         If LastResultRow < 3 Then
54             Application.ScreenUpdating = True
55             Exit Sub
56         End If
57         If LastResultRow < 4 Then GoTo NoSort 'No Sort on a single item
58         'Sort By ID
59         With .Sort

```

1 2 3


```

58 | 1 2 3 | .SortFields.Clear
59 | | .SortFields.Add Key:=TransDB.Range("Q3" ), SortOn:=xlSortOnValues, Order:=
60 | | xlAscending, DataOption:=xlSortNormal 'Sort
61 | | .SetRange TransDB.Range("Q3:X" & LastResultRow) 'Set Range
62 | | .Apply 'Apply Sort
63 | | End With
64 | | NoSort:
65 | | Transactions.Range("D4:K" & LastResultRow + 1).Value = .Range("Q3:X" &
66 | | LastResultRow).Value
67 | | Transactions.Range("F2" ).Select
68 | | Application.ScreenUpdating = True
69 | | End With
70 | | End Sub
71 | Sub Transaction_SaveUpdate()
72 | | 'Check For Required field
73 | | With TransactFrm
74 | | | If .Field1.Value = Empty Then
75 | | | | MsgBox "Please make sure to add a Transaction Name before saving"
76 | | | | Exit Sub
77 | | | End If
78 | | | If Transactions.Range("B4" ).Value = Empty Then 'New Transaction
79 | | | | TransRow = TransDB.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
80 | | | | TransDB.Range("A" & TransRow).Value = Transactions.Range("B5" ).Value 'Next
81 | | | | Transaction ID
82 | | | | TransDB.Range("H" & TransRow).Value = "=Row()"
83 | | | Else 'Existing Trans.
84 | | | | TransRow = Transactions.Range("B4" ).Value 'Transaction Row
85 | | | End If
86 | | | For TransCol = 2 To 7
87 | | | | Set TransFld = .Controls("Field" & TransCol - 1)
88 | | | | TransDB.Cells(TransRow, TransCol).Value = TransFld.Value
89 | | | Next TransCol
90 | | End With
91 | | Unload TransactFrm
92 | | Transaction_RefreshList
93 | End Sub

```

```

1 Option Explicit
2 Sub CreateTreeView()
3     Dim OrdRow As Long, LastRow As Long, LastResultRow As Long, CustRow As Long
4     Dim OrdType As String, CustName As String, OrdID As String
5     Dim OrderSht As Worksheet
6
7     Set OrderSht = ThisWorkbook.Sheets("Order List" )
8     With OrderDB
9         LastRow = .Range("A99999" ).End(xlUp).Row
10        If LastRow < 4 Then Exit Sub
11        OrdersFrm.TreeView1.Nodes.Clear 'Clear all nodes
12        OrdersFrm.Show
13        'Get Unique Customers (Parents)
14        On Error Resume Next
15        .Names("Criteria" ).Delete
16        On Error GoTo 0
17        Application.ScreenUpdating = False
18        Application.Calculation = xlCalculationManual
19        .Range("C3:C" & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
20            "W2:W2" ), Unique:=True
21        LastResultRow = .Range("W99999" ).End(xlUp).Row
22        If LastResultRow < 3 Then GoTo NoData
23        For CustRow = 3 To LastResultRow
24            With OrdersFrm.TreeView1.Nodes
25                .Add Key:=OrderSht.Range("W" & CustRow).Value, Text:=OrderSht.Range("W" &
26                    CustRow).Value 'Add Customer
27                If OrdersFrm.IncW0.Value = True Then .Add Relative:=OrderSht.Range("W" &
28                    CustRow).Value, Relationship:=tvwChild, Key:=OrderSht.Range("W" & CustRow)
29                    .Value & "-Work Orders" , Text:="Work Orders" , Image:="Work Orders"
30                If OrdersFrm.IncInvoice.Value = True Then .Add Relative:=OrderSht.Range(
31                    "W" & CustRow).Value, Relationship:=tvwChild, Key:=OrderSht.Range("W" &
32                    CustRow).Value & "-Invoices" , Text:="Invoices" , Image:="Invoices"
33                If OrdersFrm.IncPurch.Value = True Then .Add Relative:=OrderSht.Range("W"
34                    & CustRow).Value, Relationship:=tvwChild, Key:=OrderSht.Range("W" &
35                    CustRow).Value & "-Purchases" , Text:="Purchases" , Image:="Purchases"
36            End With
37        Next CustRow
38
39        For OrdRow = 4 To LastRow
40            OrdType = Application.WorksheetFunction.Proper(.Range("B" & OrdRow).Value) &
41                "s" 'Order Type
42            If OrdType = "Invoices" And OrdersFrm.IncInvoice.Value = False Then GoTo
43                NextOrder
44            If OrdType = "Purchases" And OrdersFrm.IncPurch.Value = False Then GoTo
45                NextOrder
46            If OrdType = "Work Orders" And OrdersFrm.IncW0.Value = False Then GoTo
47                NextOrder
48            CustName = .Range("C" & OrdRow).Value 'Customer Name
49            OrdID = .Range("A" & OrdRow).Value 'Order ID
50            With OrdersFrm.TreeView1.Nodes
51                .Add Relative:=CustName & "-" & OrdType, Relationship:=tvwChild, Key:=
52                    "Order" & OrdID, _
53                    Text:=Left(OrdType, Len(OrdType) - 1) & " #" & OrdID 'Add Order
54            End With
55            NextOrder:
56        Next OrdRow
57    End With
58    NoData:
59    Application.Calculation = xlCalculationAutomatic
60    Application.ScreenUpdating = True

```

```
49 1 End Sub
50
51 Sub PrintSelectedOrdersOld()
52     Dim OrdCount As Long
53     Dim OrdID As String
54     Dim Node As Node
55     With OrdersFrm
56         If .EnableCheckboxes.Value = False Then
57             MsgBox "Please make sure to select the 'Enable Checkboxes, then select orders
58                 to be printed"
59             Exit Sub
60         End If
61         For Each Node In .TreeView1.Nodes
62             If InStr(Node.Text, "#" ) > 0 And Node.Checked = True Then
63                 OrdCount = OrdCount + 1
64                 OrdID = Replace(Node.Key, "Order" , "" ) 'Extract Order ID
65                 Orders.Range("Q5" ).Value = OrdID
66                 Order_Print
67             End If
68         Next Node
69     End With
70 End Sub
71 Sub ShowTreeView()
72     OrdersFrm.Show
73 End Sub
```

```

1 Option Explicit
2 Dim UserRow As Long, UserCol As Long, LastRow As Long, LastResultRow As Long
3 Dim UserFld As control
4
5 Sub User_AddNew()
6     Users.Range("B3,B4" ).ClearContents 'Clear Selected User Row and User DB Row
7     UserFrm.Show
8 End Sub
9
10 Sub User_AddPicture()
11     Dim UserPic As FileDialog
12     Dim PicName As String, PicPath As String, PicFolder As String
13
14     If [UserFolder] = "" Then
15         MsgBox "Please select a folder for User pictures"
16         BrowseForUserPicFolder
17         If [UserFolder] = "" Then Exit Sub
18     End If
19     PicFolder = [UserFolder] 'Set Picture Folder
20     Set UserPic = Application.FileDialog(msoFileDialogFilePicker)
21     With UserPic
22         .Title = "Select User Picture"
23         .Filters.Add "Select Picture" , "*.jpg, *.png, *.jpeg, *.gif" , 1
24         If .Show <> -1 Then GoTo NoSelection
25         PicPath = .SelectedItems(1) 'Entire Picture Path
26     End With
27     PicName = Dir(PicPath) 'Set Picture Name
28     UserFrm.Field4.Value = PicName 'Set Picture Name
29     'Copy User Picture to picture folder
30     On Error Resume Next
31     FileCopy PicPath, PicFolder & "\" & PicName
32     On Error GoTo 0
33     NoSelection:
34 End Sub
35 Sub User_ClearFilter()
36     Users.Range("F2:G2" ).ClearContents
37 End Sub
38
39 Sub User_Delete()
40     If MsgBox("Are you sure you want to delete this User?" , vbYesNo, "Delete User" ) =
41         vbNo Then Exit Sub
42     If Users.Range("B4" ).Value = Empty Then Exit Sub
43     UserRow = Users.Range("B4" ).Value 'User Row
44     For UserCol = 2 To 10
45         UserDB.Cells(UserRow, UserCol).ClearContents
46     Next UserCol
47     Users.Range("B3,B4" ).ClearContents
48     User_RefreshList
49 End Sub
50
51
52
53 Sub User_Edit()
54     With Users
55         If .Range("B4" ).Value = Empty Then
56             MsgBox "Please select on a User to edit"
57             Exit Sub
58         End If
59         UserRow = .Range("B4" ).Value 'User Row
60     With UserFrm

```

1 2 3

```

1 2 3
61 For UserCol = 2 To 5
62     Set UserFld = .Controls("Field" & UserCol - 1)
63     UserFld.Value = UserDB.Cells(UserRow, UserCol).Value 'Add User Data
64 Next UserCol
65     .Show 'Display User Form
66 End With
67 End With
68 End Sub
69 Sub User_RefreshList()
70     With UserDB
71         Users.Range("D4:H9999" ).ClearContents
72         LastRow = .Range("A9999" ).End(xlUp).Row
73         If LastRow < 4 Then Exit Sub
74         Application.ScreenUpdating = False
75         .Range("A3:F" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
76             "K2:L3" ), CopyToRange:=.Range("N2:R2" ), Unique:=True
77         LastResultRow = .Range("N9999" ).End(xlUp).Row
78         If LastResultRow < 3 Then
79             Application.ScreenUpdating = True
80             Exit Sub
81         End If
82         If LastResultRow < 4 Then GoTo NoSort 'No Sort on a single item
83         'Sort By ID
84         With .Sort
85             .SortFields.Clear
86             .SortFields.Add Key:=UserDB.Range("N3" ), SortOn:=xlSortOnValues, Order:=
87                 xlAscending, DataOption:=xlSortNormal 'Sort
88             .SetRange UserDB.Range("N3:R" & LastResultRow) 'Set Range
89             .Apply 'Apply Sort
90         End With
91         NoSort:
92         Users.Range("D4:H" & LastResultRow + 1).Value = .Range("N3:R" & LastResultRow).
93             Value
94         Users.Range("F2" ).Select
95         Application.ScreenUpdating = True
96     End With
97 End Sub
98 Sub User_SaveUpdate()
99     'Check For Required field
100     With UserFrm
101         If .Field1.Value = Empty Then
102             MsgBox "Please make sure to add a User Name before saving"
103             Exit Sub
104         End If
105         If Users.Range("B4" ).Value = Empty Then 'New User
106             UserRow = UserDB.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
107             UserDB.Range("A" & UserRow).Value = Users.Range("B5" ).Value 'Next User ID
108             UserDB.Range("F" & UserRow).Value = "=Row()"
109         Else 'Existing User.
110             UserRow = Users.Range("B4" ).Value 'User Row
111         End If
112         For UserCol = 2 To 5
113             Set UserFld = .Controls("Field" & UserCol - 1)
114             UserDB.Cells(UserRow, UserCol).Value = UserFld.Value
115         Next UserCol
116     End With
117     Unload UserFrm
118     User_RefreshList
119 End Sub

```

```

1 Option Explicit
2 Dim VendRow As Long, VendCol As Long, LastRow As Long, LastResultRow As Long
3 Dim VendFld As control
4
5 Sub Vendor_AddNew()
6     Vendors.Range("B3,B4" ).ClearContents 'Clear Selected Vendor Row and Vend DB Row
7     VendorFrm.Show
8 End Sub
9
10 Sub Vendor_ClearFilter()
11     Vendors.Range("F2:H2" ).ClearContents
12 End Sub
13
14 Sub Vendor_Delete()
15     If MsgBox("Are you sure you want to delete this Vendor?" , vbYesNo, "Delete Vendor"
16     ) = vbNo Then Exit Sub
17     If Vendors.Range("B4" ).Value = Empty Then Exit Sub
18     VendRow = Vendors.Range("B4" ).Value 'Vendor Row
19     For VendCol = 2 To 10
20         VendorDB.Cells(VendRow, VendCol).ClearContents
21     Next VendCol
22     Vendors.Shapes("EditGrp" ).Visible = msoFalse
23     Vendors.Range("B3,B4" ).ClearContents
24     Vendor_RefreshList
25 End Sub
26
27
28 Sub Vendor_Edit()
29     With Vendors
30         If .Range("B4" ).Value = Empty Then
31             MsgBox "Please select on a Vendor to edit"
32             Exit Sub
33         End If
34         VendRow = .Range("B4" ).Value 'Vendor Row
35         With VendorFrm
36             For VendCol = 2 To 8
37                 Set VendFld = .Controls("Field" & VendCol - 1)
38                 VendFld.Value = VendorDB.Cells(VendRow, VendCol).Value 'Add Vendor Data
39             Next VendCol
40             .Show 'Display User Form
41         End With
42     End With
43 End Sub
44
45
46 Sub Vendor_RefreshList()
47     With VendorDB
48         Vendors.Range("D4:L9999" ).ClearContents
49         LastRow = .Range("A99999" ).End(xlUp).Row
50         If LastRow < 4 Then Exit Sub
51         Application.ScreenUpdating = False
52         .Range("A3:I" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
53         "M2:N3" ), CopyToRange:=.Range("P2:X2" ), Unique:=True
54         LastResultRow = .Range("X99999" ).End(xlUp).Row
55         If LastResultRow < 3 Then
56             Application.ScreenUpdating = True
57             Exit Sub
58         End If
59         If LastResultRow < 4 Then GoTo NoSort 'No Sort on a single item
60         'Sort By ID

```

1 2

```

1 2
60 With .Sort
61     .SortFields.Clear
62     .SortFields.Add Key:=VendorDB.Range("P3" ), SortOn:=xlSortOnValues, Order:=
        xlAscending, DataOption:=xlSortNormal 'Sort
63     .SetRange VendorDB.Range("P3:X" & LastResultRow) 'Set Range
64     .Apply 'Apply Sort
65 End With
66 NoSort:
67     Vendors.Range("D4:L" & LastResultRow + 1).Value = .Range("P3:X" & LastResultRow
        ).Value
68     Application.ScreenUpdating = True
69 End With
70 End Sub
71 Sub Vendor_SaveUpdate()
72     'Check For Required field
73     With VendorFrm
74         If .Field1.Value = Empty Then
75             MsgBox "Please make sure to add a Vendor Name before saving"
76             Exit Sub
77         End If
78         If Vendors.Range("B4" ).Value = Empty Then 'New Vendor
79             VendRow = VendorDB.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
80             VendorDB.Range("A" & VendRow).Value = Vendors.Range("B5" ).Value 'Next
                Vendor ID
81             VendorDB.Range("I" & VendRow).Value = "=Row()"
82         Else 'Existing Vend.
83             VendRow = Vendors.Range("B4" ).Value 'Vendor Row
84         End If
85         For VendCol = 2 To 8
86             Set VendFld = .Controls("Field" & VendCol - 1)
87             VendorDB.Cells(VendRow, VendCol).Value = VendFld.Value
88         Next VendCol
89     End With
90     Unload VendorFrm
91     Vendor_RefreshList
92 End Sub

```

- , 86
- ## A
- AcctRow, 41
 Activate, 61, 63, 65, 66, 81
 ActiveSheet, 33, 50, 56, 69
 ActiveStatus, 79, 80
 ActiveWindow, 59
 Add, 35, 37, 49-51, 61, 65, 66, 68, 69, 73, 74, 76-79, 82, 85, 86, 88, 89, 91
 Address, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 56
 Admin, 40, 41, 44-46, 61, 62, 73, 79-81
 AdvancedFilter, 48, 51, 62-64, 68, 69, 73, 74, 77, 78, 82, 84, 86, 89, 90
 AllBody, 57, 58
 AllowMultiSelect, 44
 APIKey, 57, 58
 Application, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 44, 48-52, 54, 56, 57, 62-64, 67, 76, 77, 79, 82-86, 88-91
 Apply, 49, 51, 65, 68, 69, 73, 74, 77, 78, 85, 89, 91
 AppUserFrm, 54-57
- ## B
- BrowseBtn_Click, 33, 39, 40, 42
 BrowseForEmployeePicFolder, 44, 50
 BrowseForMainSharedFolder, 44, 46
 BrowseForOrderPicFolder, 44
 BrowseForPicFolder, 79
 BrowseForProductPicFolder, 44, 76
 BrowseForUserPicFolder, 44, 88
 ButRow, 57
 ButShp, 54, 56
- ## C
- Calculation, 54, 56, 57, 83, 86
 Cancel, 43
 CancelBtn_Click, 32-34, 39-43
 CancelBtn_Exit, 43
 CapPlan, 79, 80
 CapSched_Refresh, 80
 Caption, 33, 40, 55-57, 79
 Cell, 56
 CellAdd, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 Cells, 48, 49, 51, 52, 54-56, 62-64, 76, 77, 84, 85, 88-91
 CellVal, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 ChangeLanguage, 6, 54
 ChangeRow, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 Chart, 61
 ChartArea, 61
 ChartObjects, 61
 Charts, 61
 ChatGPTAPI, 57
 CheckBoxes, 35, 37, 83
 Checkboxes_Click, 37
 Checked, 83, 87
 Chr, 35, 57, 58, 70, 73, 77
 Clear, 49, 51, 61, 65, 68, 69, 73, 74, 77, 78, 82, 85, 86, 89, 91
 ClearContents, 16, 48, 50, 51, 59, 61, 62, 64, 67, 69, 73, 74, 76, 77, 79, 84, 88-90
 Close, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 CloseMode, 43
 CodeName, 54, 56
 Color, 56
 Column, 54
 CommandButton, 57
 CommandButton1_Click, 37
 control, 48, 50, 54, 57, 76, 81, 84, 88, 90
 Controls, 48, 49, 51, 52, 55-57, 76, 77, 84, 85, 89-91
 Copy, 61, 67, 70, 73
 CopyToRange, 48, 51, 62-64, 68, 69, 73, 74, 77, 78, 82, 84, 86, 89, 90
 CostUnit, 79, 80
 Count, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 61
 CountLarge, 9, 12, 17, 20, 23, 26, 29, 31
 CreateAllOrderImages, 61
 CreateObject, 57
 CreateOrderPicture, 61
 CreateTreeView, 35, 36, 86
 CriteriaRange, 48, 51, 62-64, 68, 69, 73, 74, 77, 84, 89, 90
 CurrentUser, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 CustCol, 48, 49
 CustFld, 48, 49
 CustName, 82, 86
 Customer_AddNew, 48
 Customer_ClearFilter, 48
 Customer_Delete, 48
 Customer_Edit, 48
 Customer_RefreshList, 9, 48, 49
 Customer_SaveUpdate, 32, 49
 CustomerDB, 48, 49
 CustomerFrm, 48, 49, 55
 Customers, 48, 49, 81
 CustRow, 48, 49, 82, 86
- ## D
- Dashboard, 81
 DataOption, 49, 51, 65, 68, 69, 73, 74, 77, 78, 85, 89, 91
 DBSht, 16
 Debug, 58
 DefCap, 79, 80
 Delay, 63
 Delete, 61, 65, 66, 78, 82, 86
 Designer, 55, 57
 Dir, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 36, 37, 46, 50, 59, 76, 79, 88
 DoEvents, 58, 63
- ## E
- EmplCol, 50-52
 EmplFld, 50-52
 Employee_AddNew, 50
 Employee_AddPicture, 33, 50
 Employee_AddUpdatePicture, 50, 53, 67
 Employee_ClearFilter, 50
 Employee_Delete, 51
 Employee_Edit, 51
 Employee_RefreshList, 12, 51, 52
 EmployeeDB, 50-52, 67, 73
 EmployeeFolder, 50
 EmployeeFrm, 50-53, 55
 EmployeeList_SaveUpdate, 33, 52
 EmployeePayroll_SaveUpdate, 33, 52
 Employees, 50-52, 73, 81
 EmplPic, 50
 EmplRow, 50-52, 67
 Empty, 9, 12, 16, 17, 20, 23, 26, 29, 31, 41, 46, 48-52, 56, 59, 62, 64, 65, 67, 69, 71, 74-77, 79, 80, 84, 85, 88-91
 EmpType, 52
 EnableCheckboxes, 35, 87
 EnableCheckboxes_Click, 35
 EnableEvents, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 54, 56, 57
 EntireColumn, 59
 Explicit, 6-16, 18-35, 37, 39-44, 46, 48, 50, 54, 59, 61, 67, 73, 76, 79, 81, 82, 84, 86, 88, 90
 Export, 61
- ## F
- Field1, 49, 52, 77, 85, 89, 91
 Field12, 50
 Field13, 50
 Field2, 52
 Field3, 33, 41, 52
 Field3_Change, 33, 41
 Field4, 41, 52, 76, 88
 Field5, 52
 FileCopy, 50, 76, 79, 88
 FileDialog, 44, 50, 76, 79, 88
 FileName, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46, 61
 FilePath, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46, 47
 FileText, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 Fill, 50, 59, 63, 65, 66
 FilterName, 61
 Filters, 50, 76, 79, 88
 Find, 16, 40, 41, 54, 59, 65, 74, 79, 80
 Font, 65, 66
 ForeColor, 65, 66
 FreeFile, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 FrmName, 54, 55
 FromDt, 73
- ## G
- GenerateTreeView, 37, 38, 82
 GetButtonText, 56
 GetLabelValues, 56
 GetUserformLabelsAndButtons, 57
 GroupItems, 59
 GrpShp, 59
- ## H
- HasFormula, 56
 Height, 61
 Hidden, 59
 Hide, 53, 80
 HideSelection, 35, 37
- ## I
- i, 63
 Image, 82, 86
 ImageList, 35, 37
 ImageNum, 35, 37
 InclInvoice, 35, 86
 InclInvoice_Click, 35
 InclInvoices, 82
 InclInvoices_Click, 37
 IncPurch, 35, 86
 IncPurch_Click, 35
 IncPurchases, 82
 IncPurchases_Click, 37
 IncWO, 35, 86

- IncWO_Click, 35
 IncWorkOrders, 82
 IncWorkOrders_Click, 37
 Indentation, 35, 37
 Input, 46
 InStr, 36, 37, 44, 46, 58, 59, 83, 87
 InStrRev, 58
 Interior, 56
 Intersect, 6, 8, 9, 11, 12, 14-17, 20-23, 25-31
 Invoice, 35, 37
 IRibbonControl, 81
 ItemDBCol, 61, 62
 ItemDBRow, 61, 62, 64, 67, 70, 71
 ItemRow, 61, 63, 64
- K**
- Key, 35-37, 49, 51, 65, 68, 69, 73, 74, 77, 78, 82, 83, 85-87, 89, 91
 Kill, 47
- L**
- LabCont, 54-57
 Label, 57
 LabelEdit, 35, 37
 LangCol, 54-56
 LangDB, 54, 56, 57
 LangNumb, 54
 LangRow, 54-57
 Language, 57
 LastItemRow, 61, 63, 64, 67, 70, 71
 LastResultRow, 48-51, 61-65, 67-69, 73, 74, 76, 77, 82, 84-86, 88-91
 LastRow, 48, 50, 51, 54, 55, 61-64, 67-69, 73-78, 82, 84, 86, 88-90
 LastUserRow, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30
 LblDefCap, 40, 79
 Left, 9, 12, 23, 26, 29, 31, 46, 58, 82, 86
 LeftPos, 46
 Len, 46, 58, 82, 86
 ListImages, 35, 37
 LoadPicture, 36, 37
 Location, 61
 LOF, 46
 LoggedInUser, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46, 59
 LoginFrm, 55, 59
- M**
- MainMenuSelect, 59
 Msg, 46
 MkDir, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 Model, 57, 58
 MSForms, 43, 57
 MsgBox, 20, 44, 46, 48-52, 54, 58, 59, 62, 63, 67, 69, 71, 74-77, 79, 80, 83-85, 87-91
 MsgNumb, 46
 msoCTrue, 9, 12, 23, 26, 29, 31, 50, 59, 60, 63
 msoFalse, 9, 12, 23, 26, 29, 31, 48, 50, 51, 59, 60, 63, 73, 90
 msoFileDialogFilePicker, 50, 76, 79, 88
 msoFileDialogFolderPicker, 44
 MyImgList, 35, 37
- N**
- Name, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 33, 50, 56, 57, 59, 61, 69
 Names, 78, 82, 86
 NameType, 61
 newValue, 46
- NextFile, 46
 NextOrder, 82, 86
 NextShp, 56
 NextUser, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30
 NoAdditions, 68, 71
 NoButtons, 54
 NoData, 82, 83, 86
 Node, 83, 87
 NoDeductions, 68, 70, 71
 Nodes, 82, 83, 86, 87
 Noltems, 64
 NoLabels, 54, 55
 NoPic, 59
 NoPTOLtems, 68, 69
 NoSelection, 44, 45, 50, 76, 79, 88
 NoSheetFields, 54
 NoSort, 49, 51, 68, 69, 73, 74, 77, 84, 85, 89-91
 NoTaxItems, 68, 70
 NotSaved, 62
 NotSelected, 44
 NoUserformButtons, 55, 56
 Now, 59, 61
- O**
- OkBtn_Click, 34
 OldValue, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 Open, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46, 57
 OrdCount, 83, 87
 Order, 49, 51, 65, 68, 69, 73, 74, 77, 78, 85, 89, 91
 Order_AddNew, 61, 62
 Order_Delete, 62
 Order_Load, 16, 17, 62
 Order_Print, 63, 83, 87
 Order_SavedMsg, 63, 64
 Order_SaveUpdate, 63
 OrderCol, 61-64
 OrderDB, 35, 37, 61-65, 86
 OrderItems, 62-64
 OrderPic, 36, 37
 OrderPicFolder, 36, 37, 61
 OrderRow, 61-64
 Orders, 61-66, 81, 83, 87
 Orders_List, 62, 64-66
 OrdersFrm, 55, 86, 87
 OrderSht, 61, 82, 86
 OrdersTreeView, 55, 82, 83
 OrderTab_Invoice, 63, 65, 81
 OrderTab_PO, 63, 65, 81
 OrderTab_WO, 63, 66, 81
 OrderType, 61-63
 OrdID, 82, 83, 86, 87
 OrdRow, 82, 86
 OrdType, 82, 86
 Output, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30
 oXMLHTTP, 57, 58
- P**
- PaidEmplID, 73, 75
 PaidEmpl, 67, 69, 70, 74, 75
 PaidEmpRow, 67, 69, 70, 73, 75
 PaidItemRow, 67, 70, 71, 73
 ParsedAnswer, 57, 58
 Password, 59
 Paste, 61
 PasteSpecial, 67, 70, 73
 PayEmpl_Load, 20, 67
 PayEmpl_LoadPayItems, 67
 PayEmpl_LoadTimesheetEntries, 67, 69
 PayEmpl_Previous, 69
 PayEmpl_SaveNext, 69
 PayEmpRow, 67, 73, 75
 PayFreq, 73
 PayItems, 67, 68, 70, 71
 Payroll, 50, 52, 67-69, 73-75, 81
 Payroll_AddAllEmployees, 20, 73, 74
 Payroll_AddNew, 73
 Payroll_LoadList, 20, 73, 75
 Payroll_LoadPayroll, 20, 74
 Payroll_SaveUpdate, 75
 PayrollDB, 73-75
 PayrollRow, 67, 73-75
 PayRow, 67, 68
 PicFolder, 44, 50, 59, 76, 79, 88
 PicName, 50, 59, 76, 79, 88
 PicPath, 35-37, 50, 59, 76, 79, 88
 Picture, 35-37, 79, 80
 PictureFolder, 50, 79
 Pictures, 61
 Print, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 58
 PrintBtn_Click, 35, 37
 PrintOut, 63, 75
 PrintPayStub, 75
 PrintSelectedOrders, 37, 83
 PrintSelectedOrdersOld, 35, 87
 ProdCol, 76, 77
 ProdFld, 76, 77
 ProdPic, 76
 ProdRow, 16, 76, 77
 Product_AddNew, 76
 Product_AddPicture, 39, 76
 Product_ClearFilter, 76
 Product_Delete, 76
 Product_Edit, 76
 Product_RefreshList, 23, 76-78
 Product_SaveUpdate, 39, 77
 ProductDB, 16, 76-78
 ProductFolder, 76
 ProductFrm, 55, 76, 77
 Products, 76, 77, 81
 Prompt, 57, 58
 Proper, 82, 86
 Purchase, 35, 37
- R**
- Range, 6, 8, 9, 11, 12, 14-17, 20-23, 25-31, 35, 40, 41, 44-46, 48-52, 54-57, 59, 61-80, 82-91
 RateLbl, 33
 RecRow, 16
 Relationship, 82, 86
 Relative, 82, 86
 Replace, 46, 58, 83, 87
 Reset, 57
 ResName, 79, 80
 Resource_AddNew, 79
 Resource_AddPicture, 40, 79
 Resource_Edit, 79
 Resource_SaveUpdate, 40, 80
 ResourceFrm, 55, 79, 80
 Resources, 79, 80
 ResPic, 79
 Response, 57, 58
 responseText, 58
 ResRow, 79, 80
 RestartFrms, 54
 ResType, 40, 79, 80
 ResType_Change, 40

- ResultRow, 61-63, 67, 68, 73
 ReturnBoolean, 43
 RGB, 56, 65, 66
 Right, 46, 58
 Role, 46
 Row, 8, 9, 11, 12, 14-17, 20-23, 25-31, 40, 41, 48, 49, 51, 52, 54-56, 59, 61-65, 67-71, 73-75, 77-80, 82, 84-86, 89-91
- ## S
- SalesUnit, 79, 80
 SaveBtn_Click, 32, 33, 39-43
 ScreenUpdating, 48, 49, 51, 52, 54, 56, 57, 62-64, 67, 77, 83-86, 89-91
 ScrollColumn, 59
 SelectedItem, 35-37
 SelectedItems, 44, 45, 50, 76, 79, 88
 Selection, 61
 SellItem, 35-37
 SelLang, 54
 SelLanguage, 54
 SelRow, 61, 65, 67, 69-74, 79
 Send, 58
 SentFrom, 46
 SetRange, 49, 51, 65, 68, 69, 73, 74, 77, 78, 85, 89, 91
 SetRequestHeader, 58
 SettingImageList, 35, 36
 SettingImages, 37, 38
 Shape, 54, 56, 59, 61
 Shapes, 9, 12, 23, 26, 29, 31, 48, 50, 51, 54, 56, 59-61, 63, 65, 66, 73, 90
 SharedFold, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 SharedFolder, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 44, 46
 SharedFolder_Check, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 ShareSyncEnabled, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30
 ShareSyncRun, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46, 47
 SheetNm, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 Sheets, 46, 54, 61, 82, 86
 Show, 44, 48, 50, 51, 57, 59, 76, 77, 79, 80, 83, 84, 86-90
 ShowTrainTreeView, 83
 ShowTreeView, 87
 ShowVendForm, 57
 ShtCdNm, 54
 SkipSort, 65, 69
 Sort, 49, 51, 65, 68, 69, 73, 74, 77, 78, 84, 89, 91
 SortFields, 49, 51, 65, 68, 69, 73, 74, 77, 78, 85, 89, 91
 SortOn, 49, 51, 65, 68, 69, 73, 74, 77, 78, 85, 89, 91
 Split, 46
 StartString, 57
 StartTime, 63
 StoreMgmt, 59
 StringLength, 57
- ## T
- Target, 6, 8, 9, 11, 12, 14-17, 20-23, 25-31
 Temp, 57, 58
 TempChart, 61
 TempShp, 61
 Text, 35, 37, 54, 56, 82, 83, 86, 87
 TextArr, 46
 TextFile, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46
 TextFrame2, 54, 56, 65, 66
 TextRange, 54, 56, 65, 66
 ThisWorkbook, 46, 54, 55, 57, 61, 82, 86
 Timer, 63
 Timesheet, 69
 Title, 44, 50, 76, 79, 88
 ToDt, 73
 Toolbar_Admin, 81
 Toolbar_Customers, 81
 Toolbar_Dashboard, 81
 Toolbar_Employees, 81
 Toolbar_Invoice, 81
 Toolbar_Payroll, 81
 Toolbar_Products, 81
 Toolbar_PurchaseOrders, 81
 Toolbar_Transactions, 81
 Toolbar_Users, 81
 Toolbar_Vendors, 81
 Toolbar_WorkOrders, 81
 Top, 9, 12, 23, 26, 29, 31
 TopPos, 46
 TransactFrm, 55, 56, 84, 85
 Transaction_AddNew, 84
 Transaction_ClearFilter, 84
 Transaction_Delete, 84
 Transaction_Edit, 84
 Transaction_RefreshList, 26, 84, 85
 Transaction_SaveUpdate, 41, 85
 Transactions, 81, 84, 85
 TransCol, 84, 85
 TransDB, 84, 85
 TransFld, 84, 85
 Translate, 57, 58
 Transparency, 63
 TransRow, 84, 85
 TreeView1, 35, 36, 86, 87
 TreeView1_Click, 35
 TreeView2, 37, 82, 83
 TreeView2_Click, 37
 Trim, 58
 tvwChild, 82, 86
 tvwManual, 35, 37
 TypeRow, 40, 52, 79, 80
- ## U
- Undo, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30
 Unique, 48, 51, 62-64, 68, 69, 73, 74, 77, 78, 82, 84, 86, 89, 90
 Unload, 32-34, 39-43, 49, 52, 53, 59, 77, 80, 85, 89, 91
 UpdateOrdersBtn_Click, 36
 UsedRange, 56
 User_AddNew, 88
 User_AddPicture, 42, 88
 User_ClearFilter, 88
 User_Delete, 88
 User_Edit, 88
 User_RefreshList, 29, 88, 89
 User_SaveUpdate, 42, 89
 UserCol, 88, 89
 UserDB, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 59, 88, 89
 UserFld, 88, 89
 UserFolder, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46, 59, 88
 UserForm_Activate, 36, 38
 UserForm_QueryClose, 43
 UserFrm, 55, 56, 88, 89
 UserLogin, 34, 59
 UserLogin_Show, 59
 UserLogout, 59
 Username, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 46, 59
 UserPic, 88
 UserPicture, 50, 59
 UserPW, 59
 UserRow, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 59, 88, 89
 Users, 81, 88, 89
- ## V
- Validation, 65, 66
 Value, 8, 9, 11, 12, 14-17, 20-23, 25-31, 33, 35, 37, 40, 41, 44-46, 48-52, 54-57, 59, 61-71, 73-77, 79, 80, 82-91
 VBComponents, 55, 57
 vbCrLf, 58
 vbDirectory, 8, 11, 14, 15, 21, 22, 25, 27, 28, 30, 36, 37, 46, 50, 59
 vbNewLine, 46, 58
 vbNo, 48, 51, 62, 76, 84, 88, 90
 VBProject, 55, 57
 vbYesNo, 48, 51, 62, 76, 84, 88, 90
 VendCol, 90, 91
 VendFld, 90, 91
 Vendor_AddNew, 90
 Vendor_ClearFilter, 90
 Vendor_Delete, 90
 Vendor_Edit, 90
 Vendor_RefreshList, 31, 90, 91
 Vendor_SaveUpdate, 43, 91
 VendorDB, 90, 91
 VendorFrm, 43, 55-57, 90, 91
 Vendors, 81, 90, 91
 VendRow, 90, 91
 Visible, 9, 12, 23, 26, 29, 31, 48, 50, 51, 59, 60, 63, 73, 90
- ## W
- Where, 61
 Width, 61
 WkSht, 54
 WordToTranslate, 57
 Work_Order, 35, 37
 Workbook_CheckAndSync, 46
 Worksheet, 8, 11, 14-16, 21, 22, 25, 27, 28, 30, 54, 61, 82, 86
 Worksheet_Change, 6, 8, 9, 11, 12, 14-16, 20-23, 25-31
 Worksheet_SelectionChange, 9, 12, 17, 20, 23, 26, 29, 31
 WorksheetFunction, 82, 86
 Worksheets, 16
- ## X
- xlAscending, 49, 51, 69, 73, 74, 77, 78, 85, 89, 91
 xlCalculationAutomatic, 56, 57, 83, 86
 xlCalculationManual, 54, 86
 xlDescending, 65, 68, 74
 xlFilterCopy, 48, 51, 62-64, 68, 69, 73, 74, 77, 78, 82, 84, 86, 89, 90
 xlFormulas, 65, 74
 xlLocationAsObject, 61
 xlPasteFormulas, 67, 70, 73
 xlSortNormal, 49, 51, 65, 68, 69, 73, 74, 77, 78, 85, 89, 91

xlSortOnValues, [49](#), [51](#), [65](#), [68](#), [69](#), [73](#), [74](#), [77](#),
[78](#), [85](#), [89](#), [91](#)

xlUp, [8](#), [11](#), [14](#), [15](#), [21](#), [22](#), [25](#), [27](#), [28](#), [30](#), [48](#),
[49](#), [51](#), [52](#), [54-56](#), [61-64](#), [67-71](#), [73-75](#), [77](#),
[78](#), [80](#), [82](#), [84-86](#), [89-91](#)

xlValidateList, [65](#), [66](#)

xlValues, [16](#), [40](#), [41](#), [54](#), [59](#), [79](#), [80](#)

xlWhole, [16](#), [40](#), [41](#), [54](#), [59](#), [65](#), [74](#), [79](#), [80](#)

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