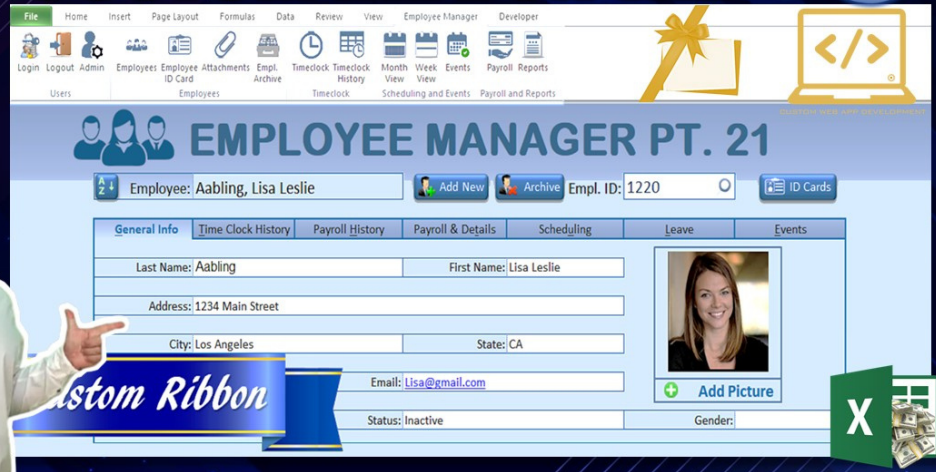


# VBA SOURCE CODE BOOK

## EMPLOYEE MANAGER PT. 21 CUSTOM RIBBON TAB



*Excel For Freelancers*

Create Your Own Excel  
Custom Ribbon Tab From  
Scratch  
[Employee Manager Pt. 21]



[DOWNLOAD  
APPLICATION](#)



[WATCH ENTIRE  
SERIES](#)



[VIEW  
TRAINING](#)

*by: Randy Austin*

# ABOUT THE AUTHOR

A two-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With nearly 300,000 YouTube subscribers, 14,000,000 video views, 200+ comprehensive training videos, and a thriving 40,000 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](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



**Microsoft®**  
Most Valuable  
Professional



# OUR COURSES & PRODUCTS



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

[Click here to learn more](#)



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

[Click here to learn more](#)



Incredible Package of 175 of my BEST Applications into a SINGLE ZIP File which also includes the "175 Workbook Library".

[Click here to learn more](#)



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

[Click here to learn more](#)

Projects.....	9
VBAProject .....	9
Documents .....	9
Sheet1.....	9
(Declarations).....	9
Worksheet_Activate [Sub ].....	9
Worksheet_Change [Sub ].....	9
Worksheet_Deactivate [Sub ].....	11
Worksheet_SelectionChange [Sub ].....	11
Sheet10.....	14
(Declarations).....	14
Sheet11.....	16
(Declarations).....	16
Worksheet_Activate [Sub ].....	16
Worksheet_Change [Sub ].....	16
Sheet12.....	18
Worksheet_Activate [Sub ].....	18
Worksheet_Change [Sub ].....	18
Sheet13.....	20
(Declarations).....	20
Worksheet_Activate [Sub ].....	20
Worksheet_Change [Sub ].....	20
Sheet14.....	22
(Declarations).....	22
Worksheet_Activate [Sub ].....	22
Sheet15.....	24
(Declarations).....	24
Worksheet_Activate [Sub ].....	24
Worksheet_Change [Sub ].....	24
Worksheet_SelectionChange [Sub ].....	24
Sheet16.....	26
(Declarations).....	26
Worksheet_Activate [Sub ].....	26
Worksheet_Change [Sub ].....	26
Worksheet_SelectionChange [Sub ].....	26
Sheet17.....	29
(Declarations).....	29
Worksheet_Activate [Sub ].....	29
Worksheet_Change [Sub ].....	29
Sheet18.....	31
(Declarations).....	31
Worksheet_Activate [Sub ].....	31
Worksheet_Change [Sub ].....	31
Sheet19.....	33
(Declarations).....	33
Worksheet_Activate [Sub ].....	33
Worksheet_Change [Sub ].....	33
Sheet2.....	35
(Declarations).....	35
Worksheet_Activate [Sub ].....	35
Worksheet_Change [Sub ].....	35
Sheet20.....	37
(Declarations).....	37
Worksheet_Activate [Sub ].....	37
Worksheet_Change [Sub ].....	37
Sheet3.....	39
(Declarations).....	39
Worksheet_Activate [Sub ].....	39
Worksheet_Change [Sub ].....	39
Sheet4.....	41
(Declarations).....	41
Worksheet_Activate [Sub ].....	41
Worksheet_Change [Sub ].....	41
Worksheet_SelectionChange [Sub ].....	42
Sheet5.....	45
(Declarations).....	45

Worksheet_Activate [Sub ].....	45
Worksheet_Change [Sub ].....	45
Worksheet_Deactivate [Sub ].....	46
Worksheet_SelectionChange [Sub ] .....	46
Sheet6.....	49
(Declarations).....	49
Worksheet_Activate [Sub ].....	49
Worksheet_Change [Sub ].....	49
Worksheet_SelectionChange [Sub ] .....	49
Sheet7.....	51
(Declarations).....	51
Worksheet_Activate [Sub ].....	51
Worksheet_Change [Sub ].....	51
Sheet8.....	53
(Declarations).....	53
Worksheet_Activate [Sub ].....	53
Worksheet_Change [Sub ].....	53
Worksheet_SelectionChange [Sub ] .....	53
Sheet9.....	55
(Declarations).....	55
Worksheet_Activate [Sub ].....	55
Worksheet_Change [Sub ].....	55
ThisWorkbook.....	57
(Declarations).....	57
Workbook_Activate [Sub ].....	57
Workbook_BeforeClose [Sub ].....	57
Workbook_Deactivate [Sub ].....	57
Workbook_SheetBeforeRightClick [Sub ].....	57
Forms.....	59
CreatePayPeriod .....	59
(Declarations).....	59
CancelBtn_Click [Sub ] .....	59
ClsBtn_Click [Sub ] .....	59
CreateBtn_Click [Sub ].....	59
EditTimeClock .....	61
(Declarations).....	61
BreakIn_AfterUpdate [Sub ] .....	61
BreakInNow_MouseDown [Sub ].....	61
BreakOut_AfterUpdate [Sub ].....	61
BreakOutNow_MouseDown [Sub ].....	61
CancelBtn_MouseDown [Sub ].....	61
ClockIn_AfterUpdate [Sub ].....	61
ClockInNow_MouseDown [Sub ] .....	61
ClockOut_AfterUpdate [Sub ].....	61
ClockOutNow_MouseDown [Sub ] .....	61
OkBtn_Click [Sub ].....	61
UserForm_Activate [Sub ] .....	62
LoginForm.....	64
(Declarations).....	64
CloseBtn_Click [Sub ] .....	64
OkBtn_Click [Sub ].....	64
PasswordField_Exit [Sub ] .....	64
UserForm_Initialize [Sub ].....	64
UserNameField_Exit [Sub ] .....	64
Modules.....	66
Admin_EventNotif.....	66
(Declarations).....	66
Notif_AddEmailAttach [Sub ] .....	66
Notif_AddNew [Sub ].....	66
Notif_CancelNew [Sub ] .....	66
Notif_Delete [Sub ].....	66
Notif_Load [Sub ] .....	66
Notif_Save [Sub ].....	66
Admin_Macros .....	69
(Declarations).....	69
Admin_SetTotalFormat [Sub ] .....	69

BrowseMainApplicationFolder [Sub ]	69
CheckMainAppFolder [Sub ]	69
CreatePayPeriods [Sub ]	69
ShowPayPeriodForm [Sub ]	70
Admin_Tabs	72
(Declarations)	72
Admin_SwitchHorizontalTabs [Sub ]	72
Application_Sync	74
(Declarations)	74
SetSyncFrequency [Sub ]	74
UpdateLocal [Sub ]	74
AutoTab_Macros	77
do_nothing [Sub ]	77
GetColLtr [Function ]	77
GetTabOrder [Function ]	77
SetOnkey [Sub ]	77
TabRange [Sub ]	78
UpOrDownArrow [Sub ]	78
CodeReset_Macs	81
(Declarations)	81
ResetCalc [Sub ]	81
StopCalc [Sub ]	81
CodeRunTime	83
(Declarations)	83
CalculateRunTime_Minutes [Sub ]	83
StartClock [Sub ]	83
StopClock [Sub ]	83
Empl_Attach	85
(Declarations)	85
Attach_Add [Sub ]	85
Attach_Delete [Sub ]	86
Attach_DisplayThumbnail [Sub ]	86
Attach_Filter [Sub ]	87
Attach_FilterClear [Sub ]	87
Attach_Open [Sub ]	88
Attach_Refresh [Sub ]	88
Attach_ShowAll [Sub ]	88
Empl_IDCard	90
(Declarations)	90
EmplIDCard_CreateTextBox [Sub ]	90
EmplIDCard_Customize [Sub ]	90
EmplIDCard_FieldChange [Sub ]	90
EmplIDCard_Hide [Sub ]	90
EmplIDCard_HideCustom [Sub ]	90
EmplIDCard_HidePic [Sub ]	91
EmplIDCard_Print [Sub ]	91
EmplIDCard_Show [Sub ]	91
EmplIDCard_ShowPic [Sub ]	91
Empl_LoadMacros	93
(Declarations)	93
Empl_EventsRefresh [Sub ]	93
Empl_LeaveRefresh [Sub ]	93
Empl_Load [Sub ]	93
Empl_PayItemsRefresh [Sub ]	94
Empl_TimeclockRefresh [Sub ]	95
Empl_WeeklySchedRefresh [Sub ]	95
Empl_Misc	98
(Declarations)	98
Empl_Archive [Sub ]	98
Empl_CancelNew [Sub ]	98
Empl_New [Sub ]	98
Empl_ReActivate [Sub ]	99
Empl_SaveNew [Sub ]	100
Empl_SortByName [Sub ]	100
Empl_PicsMacs	102
(Declarations)	102



Add_EmplPic [Sub ] .....	102
Show_EmplPic [Sub ] .....	102
Empl_Tabs .....	105
(Declarations).....	105
Empl_SwitchHorizontalTabs [Sub ] .....	105
EmpTab_Events [Sub ] .....	105
EmpTab_GenInfo [Sub ] .....	105
EmpTab_Leave [Sub ] .....	105
EmpTab_PayDetails [Sub ] .....	105
EmpTab_PayHistory [Sub ] .....	105
EmpTab_Scheduling [Sub ] .....	105
EmpTab_TimeClock [Sub ] .....	106
SwitchVerticalTabs [Sub ] .....	106
Events_Filters .....	108
(Declarations).....	108
Events_ClearFilter [Sub ] .....	108
Events_FilterTable [Sub ] .....	108
Events_Misc .....	110
(Declarations).....	110
DeleteFilters [Sub ] .....	110
Event_AddEmailAttach [Sub ] .....	110
Event_AddNew [Sub ] .....	110
Event_AddNotifDefaults [Sub ] .....	110
Event_AllDay [Sub ] .....	110
Event_CancelNew [Sub ] .....	110
Event_Delete [Sub ] .....	110
Event_Load [Sub ] .....	111
Event_Refresh [Sub ] .....	111
Event_SaveNew [Sub ] .....	112
FadeOutMessage .....	114
(Declarations).....	114
EmployIDNotFound [Sub ] .....	114
FolderPickMac .....	116
(Declarations).....	116
GetFolder [Sub ] .....	116
MailToSelected [Sub ] .....	116
Notification_Macros .....	118
(Declarations).....	118
Notification_Close [Sub ] .....	118
Notifications_Check [Sub ] .....	118
Notifications_Create [Sub ] .....	118
testting [Sub ] .....	120
Payroll_Load .....	122
(Declarations).....	122
Payroll_Load_Review [Sub ] .....	122
Payroll_Load_Summary [Sub ] .....	122
Payroll_ReviewEmpMacs .....	125
(Declarations).....	125
Payroll_LoadEmpReview [Sub ] .....	125
Payroll_Review_EditTimeClockEntry [Sub ] .....	127
Payroll_ShowEditEmployeeIcon [Sub ] .....	128
Payroll_ShowEditTimeClockIcon [Sub ] .....	128
Payroll_ViewEmployeeInfo [Sub ] .....	128
Payroll_SaveEmplPayMac .....	130
(Declarations).....	130
Payroll_PrevEmployee [Sub ] .....	130
Payroll_SaveEmployeePay [Sub ] .....	130
Payroll_SelectSections .....	136
(Declarations).....	136
Payroll_Details_To_Review [Sub ] .....	136
Payroll_SelectDetailsSection [Sub ] .....	136
Payroll_SelectReviewSection [Sub ] .....	136
Payroll_SelectSummarySection [Sub ] .....	137
Payroll_Setup .....	139
(Declarations).....	139
Payroll_AddSelectedEmpl [Sub ] .....	139

**Table of Contents**

Payroll_EmpToPay_SelectAll [Sub ].....	139
Payroll_FilterEmployees [Sub ] .....	139
Payroll_RemoveSelectedEmpl [Sub ].....	140
Payroll_SelToPay_SelectAll [Sub ].....	140
Payroll_Summary .....	143
(Declarations).....	143
Payroll_ConfirmCreate [Sub ].....	143
Payroll_LoadPayrollSummary [Sub ] .....	143
Payroll_PrintPayStubs [Sub ] .....	144
PopUp_Calendar.....	146
(Declarations).....	146
CalCol1 [Sub ].....	146
CalCol2 [Sub ].....	146
CalCol3 [Sub ].....	146
CalCol4 [Sub ].....	146
CalCol5 [Sub ].....	146
CalCol6 [Sub ].....	146
CalCol7 [Sub ].....	146
CalCol8 [Sub ].....	146
CalCol9 [Sub ].....	147
CalendarHide [Sub ].....	147
CalendarShow [Sub ] .....	147
CalFormulaReplacement [Sub ] .....	148
CheckForSheet [Sub ].....	148
CreateCalSht [Sub ] .....	148
DayBtn1 [Sub ].....	150
DayBtn10 [Sub ].....	150
DayBtn11 [Sub ].....	151
DayBtn12 [Sub ].....	151
DayBtn13 [Sub ].....	151
DayBtn14 [Sub ].....	151
DayBtn15 [Sub ].....	151
DayBtn16 [Sub ].....	151
DayBtn17 [Sub ].....	151
DayBtn18 [Sub ].....	152
DayBtn19 [Sub ].....	152
DayBtn2 [Sub ].....	152
DayBtn20 [Sub ].....	152
DayBtn21 [Sub ].....	152
DayBtn22 [Sub ].....	152
DayBtn23 [Sub ].....	152
DayBtn24 [Sub ].....	153
DayBtn25 [Sub ].....	153
DayBtn26 [Sub ].....	153
DayBtn27 [Sub ].....	153
DayBtn28 [Sub ].....	153
DayBtn29 [Sub ].....	153
DayBtn3 [Sub ].....	153
DayBtn30 [Sub ].....	154
DayBtn31 [Sub ].....	154
DayBtn32 [Sub ].....	154
DayBtn33 [Sub ].....	154
DayBtn34 [Sub ].....	154
DayBtn35 [Sub ].....	154
DayBtn36 [Sub ].....	155
DayBtn37 [Sub ].....	155
DayBtn38 [Sub ].....	155
DayBtn39 [Sub ].....	155
DayBtn4 [Sub ].....	155
DayBtn40 [Sub ].....	155
DayBtn41 [Sub ].....	155
DayBtn42 [Sub ].....	156
DayBtn5 [Sub ].....	156
DayBtn6 [Sub ].....	156
DayBtn7 [Sub ].....	156
DayBtn8 [Sub ].....	156



DayBtn9 [Sub ] .....	156
GroupCal [Sub ] .....	156
MacroLinkRemover [Sub ] .....	157
NextMonth [Sub ] .....	158
PrevMonth [Sub ] .....	158
ShowSettings [Sub ] .....	158
UnGroupCal [Sub ] .....	159
Reports_ClearPrint .....	161
(Declarations) .....	161
Reports_Clear [Sub ] .....	161
Reports_PrintOut [Sub ] .....	161
Reports_EarningStmnt .....	163
(Declarations) .....	163
Reports_EarningsStatement [Sub ] .....	163
Reports_TimeclockRpt .....	167
(Declarations) .....	167
Report_TimelockReport [Sub ] .....	167
RibbonControl_Macros .....	169
(Declarations) .....	169
GoTo_Admin [Sub ] .....	169
GoTo_EmpArchive [Sub ] .....	169
GoTo_EmpAttach [Sub ] .....	169
GoTo_EmployeeID [Sub ] .....	169
GoTo_Employees [Sub ] .....	169
GoTo_Events [Sub ] .....	169
GoTo_Login [Sub ] .....	169
GoTo_LogOut [Sub ] .....	169
GoTo_Payroll [Sub ] .....	169
GoTo_Reports [Sub ] .....	169
GoTo_ScheduleMonth [Sub ] .....	169
GoTo_ScheduleWeek [Sub ] .....	169
GoTo_TimeClock [Sub ] .....	169
GoTo_TimeClockHistory [Sub ] .....	169
RightClick_Menu .....	171
(Declarations) .....	171
CreateLeaveEvent [Sub ] .....	171
CreateLeaveEvent10 [Sub ] .....	171
CreateLeaveEvent11 [Sub ] .....	171
CreateLeaveEvent12 [Sub ] .....	171
CreateLeaveEvent13 [Sub ] .....	171
CreateLeaveEvent14 [Sub ] .....	171
CreateLeaveEvent15 [Sub ] .....	171
CreateLeaveEvent16 [Sub ] .....	171
CreateLeaveEvent17 [Sub ] .....	171
CreateLeaveEvent18 [Sub ] .....	171
CreateLeaveEvent19 [Sub ] .....	172
CreateLeaveEvent20 [Sub ] .....	172
CreateLeaveEvent21 [Sub ] .....	172
CreateLeaveEvent22 [Sub ] .....	172
CreateLeaveEvent23 [Sub ] .....	172
CreateLeaveEvent7 [Sub ] .....	172
CreateLeaveEvent8 [Sub ] .....	172
CreateLeaveEvent9 [Sub ] .....	172
RightClick_CallEvents [Sub ] .....	172
RightClick_Cancel [Sub ] .....	173
Scheduling_Macs .....	175
(Declarations) .....	175
Sched_AddComment_Event [Sub ] .....	175
Sched_AddComment_Holiday [Sub ] .....	176
Sched_ClearFilters [Sub ] .....	176
Sched_GoToEvent [Sub ] .....	176
Sched_MonthView [Sub ] .....	177
Sched_NextMonth [Sub ] .....	177
Sched_NextWeek [Sub ] .....	177
Sched_PrevMonth [Sub ] .....	177
Sched_PrevWeek [Sub ] .....	178

Sched_Refresh [Sub ]	178
Sched_ScrollToRow [Sub ]	178
Sched_ShowHideCapacity [Sub ]	178
Sched_ThisMonth [Sub ]	178
Sched_ThisWeek [Sub ]	178
Sched_WeekView [Sub ]	178
Time_Picker_Macros	180
(Declarations)	180
AMPMSel [Sub ]	180
AMSel [Sub ]	180
DoneBtn [Sub ]	180
GrpHrs [Sub ]	180
GrpMin [Sub ]	180
GrpSet [Sub ]	180
HideHrSel [Sub ]	181
HideMinSel [Sub ]	181
HoursDisplay [Sub ]	181
MacroLinkRemover [Sub ]	182
MinDisplay [Sub ]	182
PMSel [Sub ]	183
ProtTP [Sub ]	183
Select1 [Sub ]	184
Select10 [Sub ]	184
Select11 [Sub ]	184
Select12 [Sub ]	184
Select2 [Sub ]	184
Select3 [Sub ]	185
Select4 [Sub ]	185
Select5 [Sub ]	185
Select6 [Sub ]	185
Select7 [Sub ]	185
Select8 [Sub ]	185
Select9 [Sub ]	185
SelectM00 [Sub ]	186
SelectM05 [Sub ]	186
SelectM10 [Sub ]	186
SelectM15 [Sub ]	186
SelectM20 [Sub ]	186
SelectM25 [Sub ]	186
SelectM30 [Sub ]	187
SelectM35 [Sub ]	187
SelectM40 [Sub ]	187
SelectM45 [Sub ]	187
SelectM50 [Sub ]	187
SelectM55 [Sub ]	187
Set10MinInt [Sub ]	187
Set15MinInt [Sub ]	188
Set5MinInt [Sub ]	188
SetHideShow [Sub ]	188
SetThm1 [Sub ]	188
SetThm2 [Sub ]	188
SetThm3 [Sub ]	188
SetThm4 [Sub ]	188
SetThm5 [Sub ]	189
TPShow [Sub ]	189
UnGrpHrs [Sub ]	189
UnGrpMin [Sub ]	189
UnGrpSet [Sub ]	190
UnProtTP [Sub ]	190
TimeClock_EditSaveEntry	192
(Declarations)	192
EditTimeClock_Calculate [Sub ]	192
EditTimeClock_Save [Sub ]	192
TimeClock_EditEntry [Sub ]	193
TimeClock_Macros	196
(Declarations)	196

Timeclock_AlreadyEntered [Sub ] .....	196
Timeclock_DisplayPicture [Sub ] .....	196
Timeclock_EnterID [Sub ] .....	196
Timeclock_IDNotFound [Sub ] .....	199
Timeclock_StartStop [Sub ] .....	200
TimeClock_RemoteSync .....	202
(Declarations).....	202
TimeClock_SyncRemoteEntries [Sub ] .....	202
User_Login_Macros .....	205
(Declarations).....	205
CheckUser [Sub ] .....	205
CloseWorkbook [Sub ] .....	206
HideBar [Sub ] .....	206
LogOut [Sub ] .....	206
ShowLoginForm [Sub ] .....	206
StartWorkbook [Sub ] .....	206
User_Rights_Macros .....	208
(Declarations).....	208
User_SetAs_Admin [Sub ] .....	208
User_SetAs_Manager [Sub ] .....	208
User_SetAs_TimeClock [Sub ] .....	208
User_SetAs_User [Sub ] .....	208

1 Option Explicit

2  
3 Private Sub Worksheet\_Activate()  
4 UpdateLocal  
5 End Sub

6  
7 Private Sub Worksheet\_Change(ByVal Target As Range)

8 Dim EmpRow As Long

9 Dim PayPeriod As Range

10 Dim FoundPayPeriod As Range

11 'On Employee Name Change

12 On Error Resume Next

13 If Not Intersect(Target, Range("F2" )) Is Nothing Then

14 If Range("F2" ).Value <> Empty And Range("B5" ).Value <> Empty Then Empl\_Load

15 End If

16  
17 'Employee ID Change

18 If Not Intersect(Target, Range("J2" )) Is Nothing And Range("B1" ).Value = False Then 'On  
change of Employee ID but NOT on Employee Load

19 If Range("J2" ).Value <> Empty And Range("B4" ).Value <> Empty Then 'Employee ID Found

20 Range("F2" ).Value = Sheet2.Range("BC" & Range("B4" ).Value).Value

21 Else: 'Employee ID Not Found, Return to original

22 Range("F2" ).Value = Sheet2.Range("BC" & Range("B5" ).Value).Value 'Add Old

Employee Name using Old Employee Name Row

EmployIDNotFound 'Run Fade Out Message Alert

23 End If

24 End If

25  
26  
27 'On Employee Information Change

28 If Not Intersect(Target, Range("E6:K24,G66:J88, E182:K182" )) Is Nothing And Range("B1" ).  
Value = False Then 'On Cell change but not on Employee load

29 If Range("B4" ).Value = Empty Or Cells(Target.Row, Target.Column + 22).Value = Empty Or  
IsNumeric(Cells(Target.Row, Target.Column + 22).Value) = False Then Exit Sub

30 EmpRow = Range("B4" ).Value 'Employee Row

31 Sheet2.Cells(EmpRow, Cells(Target.Row, Target.Column + 22).Value).Value = Target.Value

32 End If

33  
34 'On Payroll Date Range History Change

35 If Not Intersect(Target, Range("F26, F46" )) Is Nothing And Range("F" & Target.Row).Value  
<> Empty Then

36 Set PayPeriod = Sheet4.Range("PayPeriod" )

37 Sheet4.Range("209:304" ).EntireRow.Hidden = False 'Unhide Range to use the Find  
Method, then rehide

38 Set FoundPayPeriod = PayPeriod.Find(Target.Value, , xlValues, xlWhole)

39 If Not FoundPayPeriod Is Nothing Then

40 Target.Offset(0, 2).Value = Sheet4.Range("E" & FoundPayPeriod.Row).Value 'From  
Date

41 Target.Offset(0, 4).Value = Sheet4.Range("F" & FoundPayPeriod.Row).Value 'To Date

42 End If

43 Sheet4.Range("209:304" ).EntireRow.Hidden = True

44 If Target.Row = 26 Then Empl\_TimeclockRefresh 'Run Timeclock Refresh Macros

45 End If

46  
47 'On Employee Weekly Schedule Change, Update Database, but not on Employee Load

48 If Not Intersect(Target, Range("F148:J154" )) Is Nothing And Range("B1" ).Value = False  
Then

49 Dim SchedRow As Long

50 Dim WeekRow As Long

51 Dim WeekCol As Long

52 WeekRow = Target.Row

```

1 2
53 WeekCol = Target.Column
54 If Range("K145").Value = Empty Then 'New Row
55   SchedRow = Sheet13.Range("A99999").End(xlUp).Row + 1 'First Avail Row
56   If SchedRow < 4 Then SchedRow = 4
57   Sheet13.Range("A" & SchedRow).Value = Range("J2").Value 'Set Employee ID
58   Sheet13.Range("AK" & SchedRow).Value = "=Row()" 'Set Row Number formula
59   Range("K145").Value = SchedRow
60 Else:
61   SchedRow = Range("K145").Value 'Existing ScheduleRow
62 End If
63 Sheet13.Cells(SchedRow, ((WeekRow - 148) * 5) + 2 + (WeekCol - 6)).Value = Target.Value
64   'Update Weekly Schedule List
65 End If
66
67 'On Employee Leave Change, Update Database Table, but not on Employee Load
68 If Not Intersect(Target, Range("E168:H179")) Is Nothing And Range("B1").Value = False
69 Then
70   Dim LeaveRow As Long
71   Dim LeaveID As Long
72   Dim LeaveListID As Range
73   Set LeaveListID = Sheet11.Range("LeaveList_ID")
74   If Range("L" & Target.Row).Value = Empty Then 'New Employee Leave Item
75     LeaveRow = Sheet11.Range("A99999").End(xlUp).Row + 1
76     LeaveID = Application.WorksheetFunction.Max(LeaveListID) + 1
77     Range("L" & Target.Row).Value = LeaveRow
78     Sheet11.Range("A" & LeaveRow).Value = LeaveID
79     Sheet11.Range("B" & LeaveRow).Value = Range("J2").Value 'Employee ID
80     Sheet11.Range("C" & LeaveRow).Value = Range("F2").Value 'Employee Name
81     Sheet11.Range("H" & LeaveRow).Value = "=Row()" 'Set Row
82   Else:
83     LeaveRow = Range("L" & Target.Row).Value
84   End If
85   Sheet11.Cells(LeaveRow, Target.Column - 1).Value = Target.Value
86 End If
87
88 'On Change Of Employee Federal & Local Tax Deductions, but on on Employee Load
89 If Not Intersect(Target, Range("G93:I104")) Is Nothing And Range("B1").Value = False
90 Then
91   Dim PayItemRow As Long
92   If Range("K" & Target.Row).Value = Empty Then 'On New Pay Item
93     PayItemRow = Sheet17.Range("A99999").End(xlUp).Row + 1 'Set Pay ItemRow
94     Sheet17.Range("A" & PayItemRow).Value = Range("J2").Value 'Employee ID
95     Sheet17.Range("B" & PayItemRow).Value = "Tax Deduction" 'Item Type
96     Sheet17.Range("H" & PayItemRow).Value = Target.Row 'Save Employee Manager Row
97     Sheet17.Range("I" & PayItemRow).Value = "=Row()" 'Set Database Row
98     Range("K" & Target.Row).Value = PayItemRow
99   Else:
100     PayItemRow = Range("K" & Target.Row).Value 'On Existing Pay Item
101   End If
102   Sheet17.Cells(PayItemRow, Cells(90, Target.Column).Value).Value = Target.Value 'Tax
103   Item Detail
104 End If
105
106 'On Change Of Employee Additions & Deductions, but on on Employee Load
107 If Not Intersect(Target, Range("G108:I119,G128:I139")) Is Nothing And Range("B1").Value
108 = False Then
109   If Range("K" & Target.Row).Value = Empty Then 'On New Pay Item
110     PayItemRow = Sheet17.Range("A99999").End(xlUp).Row + 1 'Set Pay ItemRow
111     Sheet17.Range("A" & PayItemRow).Value = Range("J2").Value 'Employee ID
112     If Target.Row > 127 Then Sheet17.Range("B" & PayItemRow).Value = "Pay Addition"

```

1 2 3

```

1 2 3
109 Else: Sheet17.Range("B" & PayItemRow).Value = "Pay Deduction" 'Item Type
110 Sheet17.Range("H" & PayItemRow).Value = Target.Row 'Save Employee Manager Row
111 Sheet17.Range("I" & PayItemRow).Value = "=Row()" 'Set Database Row
112 Range("K" & Target.Row).Value = PayItemRow
113 Else:
114 PayItemRow = Range("K" & Target.Row).Value 'On Existing Pay Item
115 End If
116 Sheet17.Cells(PayItemRow, Cells(140, Target.Column).Value).Value = Target.Value 'Tax
117 Item Detail
118 End If
119 End Sub
120 Private Sub Worksheet_Deactivate()
121 SetOnkey (False)
122 Dim Cal As Shape
123 On Error Resume Next
124 Set Cal = Shapes("Calendar" )
125 On Error GoTo 0
126 If Cal Is Nothing Then 'Shape Deleted
127 Application.EnableEvents = False
128 On Error Resume Next
129 Application.Undo
130 On Error GoTo 0
131 Application.EnableEvents = True
132 End If
133 End Sub
134 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
135 If Target.CountLarge > 6 Then Exit Sub
136 Dim Cal As Shape 'Calendar
137 On Error Resume Next
138 Set Cal = Shapes("Calendar" )
139 On Error GoTo 0
140 If Cal Is Nothing Then 'Shape Deleted
141 Application.EnableEvents = False
142 On Error Resume Next
143 Application.Undo
144 On Error GoTo 0
145 Application.EnableEvents = True
146 End If
147 If Not Intersect(Target, Range("K16,I26,K26,I46,K46" )) Is Nothing Then 'Change this to
148 any cell (or cells) you would like to have the Pop-Up Calendar Appear
149 CheckForSheet
150 CalendarShow
151 Else:
152 CheckForSheet
153 CalendarHide
154 End If
155 If Not Intersect(Target, Range("F6:K16,F26:K26,F46:K46,G66:J75,G182:K182" )) Is Nothing
156 Then
157 SetOnkey (True)
158 Else:
159 SetOnkey (False)
160 End If
161 'Switching Horizontal Tabs in Row 4
162 If Not Intersect(Target, Range("E4:K4" )) Is Nothing Then
163 Range("B2" ).Value = Target.Column
164 Empl_SwitchHorizontalTabs
165 End If

```

```
1
166 'Switching Vertical Tabs in Column E
167 If Not Intersect(Target, Range("E66:E129" )) Is Nothing Then
168     If Target.Value = Empty Or Target.Value = "Select Option" Then Exit Sub
169     SwitchVerticalTabs
170 End If
171 If Not Intersect(Target, Range("E19:I24" )) Is Nothing Then
172     Range("B7" ).Value = Target.Row
173     If Range("J" & Target.Row).Value <> Empty Then Attach_DisplayThumbnail
174 Else: Range("B7" ).ClearContents
175 End If
176
177 'On Selection of ID Card Fields
178 If Not Intersect(Target, Range("BD7:BF24" )) Is Nothing Then EmpIDCard_FieldChange
179 If Not Intersect(Target, Range("F2" )) Is Nothing Then UpdateLocal
180 End Sub
```



**A**

Application, [10, 11](#)  
Attach\_DisplayThumbnail, [12](#)

**C**

Cal, [11](#)  
CalendarHide, [11](#)  
CalendarShow, [11](#)  
Cells, [9-11](#)  
CheckForSheet, [11](#)  
ClearContents, [12](#)  
Column, [9-11](#)  
CountLarge, [11](#)

**E**

EmpIDCard\_FieldChange, [12](#)  
Empl\_Load, [9](#)  
Empl\_SwitchHorizontalTabs, [11](#)  
Empl\_TimeclockRefresh, [9](#)  
EmployIDNotFound, [9](#)  
EmpRow, [9](#)  
Empty, [9, 10, 12](#)  
EnableEvents, [11](#)  
EntireRow, [9](#)  
Explicit, [9](#)

**F**

Find, [9](#)  
FoundPayPeriod, [9](#)

**H**

Hidden, [9](#)

**I**

Intersect, [9-12](#)  
IsNumeric, [9](#)

**L**

LeaveID, [10](#)  
LeaveListID, [10](#)  
LeaveRow, [10](#)

**M**

Max, [10](#)

**O**

Offset, [9](#)

**P**

PayItemRow, [10, 11](#)  
PayPeriod, [9](#)

**R**

Range, [9-12](#)  
Row, [9-12](#)

**S**

SchedRow, [9, 10](#)  
SetOnkey, [11](#)  
Shape, [11](#)  
Shapes, [11](#)  
Sheet11, [10](#)  
Sheet13, [10](#)  
Sheet17, [10, 11](#)  
Sheet2, [9](#)  
Sheet4, [9](#)  
SwitchVerticalTabs, [12](#)

**T**

Target, [9-12](#)

**U**

Undo, [11](#)  
UpdateLocal, [9, 12](#)

**V**

Value, [9-12](#)

**W**

WeekCol, [9, 10](#)  
WeekRow, [9, 10](#)  
Worksheet\_Activate, [9](#)  
Worksheet\_Change, [9](#)  
Worksheet\_Deactivate, [11](#)  
Worksheet\_SelectionChange, [11](#)  
WorksheetFunction, [10](#)

**X**

xlUp, [10](#)  
xlValues, [9](#)  
xlWhole, [9](#)

1 Option Explicit

2

**E**

Explicit, [14](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A5:H999999")) Is Nothing Then
9         If Sheet4.Range("B1").Value = False And Target.Count < 2 And Sheet4.Range("J17").
            Value = "On" And Sheet4.Range("ApplicationFolder").Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject")
17            UserName = Sheet14.Range("AppUserName").Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow").Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder").Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30
31
32 End Sub
33
```

## A

Address, [16](#)

## C

Close, [16](#)

Count, [16](#)

createFolder, [16](#)

CreateObject, [16](#)

CreateTextFile, [16](#)

## D

Dir, [16](#)

## E

Empty, [16](#)

Explicit, [16](#)

## F

FileName, [16](#)

fso, [16](#)

## I

Intersect, [16](#)

## L

LastUser, [16](#)

## N

Name, [16](#)

## O

oFile, [16](#)

## R

Range, [16](#)

## S

Sheet14, [16](#)

Sheet4, [16](#)

## T

Target, [16](#)

## U

UpdateLocal, [16](#)

UserName, [16](#)

UserRow, [16](#)

## V

Value, [16](#)

vbDirectory, [16](#)

## W

Worksheet, [16](#)

Worksheet\_Activate, [16](#)

Worksheet\_Change, [16](#)

WriteLine, [16](#)

```
1  
2 Private Sub Worksheet_Activate()  
3     UpdateLocal  
4 End Sub  
5  
6 Private Sub Worksheet_Change(ByVal Target As Range)  
7     If Not Intersect(Target, Range("J13" )) Is Nothing And Range("J13" ).Value <> Empty Then  
8         Timeclock_EnterID  
9 End Sub  
10
```

**E**

Empty, [18](#)

**I**

Intersect, [18](#)

**R**

Range, [18](#)

**T**

Target, [18](#)

Timeclock\_EnterID, [18](#)

**U**

UpdateLocal, [18](#)

**V**

Value, [18](#)

**W**

Worksheet\_Activate, [18](#)

Worksheet\_Change, [18](#)



```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A4:AK999999" )) Is Nothing Then
9         If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
            Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject" )
17            UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
19            For UserRow = 308 To LastUser
20                FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
21                Set oFile = fso.CreateTextFile(FileName)
22                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                Value
23                oFile.Close
24            Next UserRow
25            Set fso = Nothing
26            Set oFile = Nothing
27        End If
28    End If
29 End Sub
30
```

## A

Address, [20](#)

## C

Close, [20](#)

Count, [20](#)

CreateObject, [20](#)

CreateTextFile, [20](#)

## E

Empty, [20](#)

Explicit, [20](#)

## F

FileName, [20](#)

fso, [20](#)

## I

Intersect, [20](#)

## L

LastUser, [20](#)

## N

Name, [20](#)

## O

oFile, [20](#)

## R

Range, [20](#)

## S

Sheet14, [20](#)

Sheet4, [20](#)

## T

Target, [20](#)

## U

UpdateLocal, [20](#)

UserName, [20](#)

UserRow, [20](#)

## V

Value, [20](#)

## W

Worksheet, [20](#)

Worksheet\_Activate, [20](#)

Worksheet\_Change, [20](#)

WriteLine, [20](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7
```

**E**

Explicit, [22](#)

**U**

UpdateLocal, [22](#)

**W**

Worksheet\_Activate, [22](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("F3" )) Is Nothing Then 'Update To/From Dates on Payroll
9         Change
10        Dim PayrollRow As Long
11        If Range("B5" ).Value = Empty Then Exit Sub
12        PayrollRow = Range("B5" ).Value 'Payroll Row
13        Range("H3" ).Value = Sheet18.Range("D" & PayrollRow).Value 'From Date
14        Range("J3" ).Value = Sheet18.Range("E" & PayrollRow).Value 'To Date
15        Reports_Clear
16        If Range("B3" ).Value = 1 Then Reports_EarningsStatement
17        If Range("B3" ).Value = 2 Then Report_TimelockReport
18    End If
19    If Not Intersect(Target, Range("F4" )) Is Nothing Then 'Update To/From Dates on Pay
20        Period Change
21        Dim PayPeriodRow As Long
22        If Range("B7" ).Value = Empty Then Exit Sub
23        PayPeriodRow = Range("B7" ).Value 'Payroll Row
24        Range("H3" ).Value = Sheet4.Range("E" & PayPeriodRow).Value 'From Date
25        Range("J3" ).Value = Sheet4.Range("F" & PayPeriodRow).Value 'To Date
26        Reports_Clear
27        If Range("B3" ).Value = 1 Then Reports_EarningsStatement
28        If Range("B3" ).Value = 2 Then Report_TimelockReport
29    End If
30    'On Employee Change
31    If Not Intersect(Target, Range("H4" )) Is Nothing Then
32        Reports_Clear
33        If Range("B3" ).Value = 1 Then Reports_EarningsStatement
34        If Range("B3" ).Value = 2 Then Report_TimelockReport
35    End If
36 End Sub
37
38 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
39     If Target.CountLarge > 1 Then Exit Sub
40     If Not Intersect(Target, Range("C6:C7" )) Is Nothing Then
41         Range("B4" ).Value = Target.Row
42         Reports_Clear
43         If Target.Row = 6 Then Reports_EarningsStatement
44         If Target.Row = 7 Then Report_TimelockReport
45     End If
46 End Sub
47
```

## C

CountLarge, [24](#)

## E

Empty, [24](#)

Explicit, [24](#)

## I

Intersect, [24](#)

## P

PayPeriodRow, [24](#)

PayrollRow, [24](#)

## R

Range, [24](#)

Report\_TimelockReport, [24](#)

Reports\_Clear, [24](#)

Reports\_EarningsStatement, [24](#)

Row, [24](#)

## S

Sheet18, [24](#)

Sheet4, [24](#)

## T

Target, [24](#)

## U

UpdateLocal, [24](#)

## V

Value, [24](#)

## W

Worksheet\_Activate, [24](#)

Worksheet\_Change, [24](#)

Worksheet\_SelectionChange, [24](#)

```

1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Target.CountLarge > 1 Then Exit Sub
9     If Not Intersect(Target, Range("F5" )) Is Nothing And Range("B4" ).Value <> Empty Then
10        'On Change of Pay Period and only on Pay Period Match
11        Range("J6" ).Value = Sheet4.Range("E" & Range("B4" ).Value + 208).Value 'From Pay
12        Date
13        Range("J8" ).Value = Sheet4.Range("F" & Range("B4" ).Value + 208).Value 'To Pay Date
14    End If
15
16    'On change of Select Payroll in Review Section, but not on Payroll Load
17    If Not Intersect(Target, Range("AL3" )) Is Nothing And Range("B1" ).Value = False Then
18        Payroll_Load_Review
19
20    'On change of Select Payroll in Summary Section, but not on Payroll Load
21    If Not Intersect(Target, Range("BK3" )) Is Nothing And Range("B1" ).Value = False Then
22        Payroll_Load_Summary
23
24    If Not Intersect(Target, Range("F7,F8,H7,H8" )) Is Nothing Then Payroll_FilterEmployees
25    'Run Payroll Filter on Change
26    'On Change of Payroll Date, Save Is On Existing Payroll, but not on Payroll Load
27    If Not Intersect(Target, Range("AM3,BD6" )) Is Nothing And Range("B1" ).Value = False And
28    Range("B3" ).Value <> Empty Then
29        Sheet18.Range("C" & Range("B3" ).Value).Value = Target.Value
30    End If
31
32 End Sub
33
34 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
35     If Target.CountLarge > 1 Then Exit Sub
36     Dim Cal As Shape 'Calendar
37     On Error Resume Next
38     Set Cal = Shapes("Calendar" )
39     On Error GoTo 0
40     If Cal Is Nothing Then 'Shape Deleted
41         Application.EnableEvents = False
42         On Error Resume Next
43         Application.Undo
44         On Error GoTo 0
45         Application.EnableEvents = True
46     End If
47
48     If Not Intersect(Target, Range("J6,J8,AM3,BD6" )) Is Nothing Then 'Change this to any
49     cell (or cells) you would like to have the Pop-Up Calendar Appear
50     CheckForSheet
51     CalendarShow
52 Else:
53     CheckForSheet
54     CalendarHide
55 End If
56
57 'On Selection of Employees To Add
58 On Error Resume Next
59 If Not Intersect(Target, Range("D11:D310,L11:L310" )) Is Nothing And Cells(Target.Row,
60 Target.Column + 1).Value <> Empty Then
61     If Target.Value = "p" Then
62         Target.Value = ""

```

1 2 3



```
54      1 2 3
      If Target.Column = 4 Then Range("B7" ).Value = False Else: Range("B8" ).Value =
55      False 'Select All option set to False
56      End
57      Else: Target.Value = "p"
58      End If
59      End If
60      'On Selection of Employees To Review
61      If Not Intersect(Target, Range("AA9:AC310" )) Is Nothing And Range("AB" & Target.Row).
62      Value <> Empty Then
63      Range("B11" ).Value = Range("AB" & Target.Row).Value 'Set Employee ID To Review
64      Range("B12" ).Value = Target.Row 'Set Row To Review
65      Payroll_ShowEditEmployeeIcon
66      Payroll_LoadEmpReview 'Run Employee Refresh
67      Else:
68      Shapes("EditEmployeeBtn" ).Visible = msoFalse
69      End If
70      'On Selection of Time Clock Entry To Review
71      If Not Intersect(Target, Range("AE14:AJ44" )) Is Nothing And Range("AE" & Target.Row).
72      Value <> Empty Then
73      Range("B14" ).Value = Target.Row
74      Payroll_ShowEditTimeClockIcon
75      Else:
76      Range("B14" ).ClearContents 'Clear selected Row
77      Shapes("EditTimeClockBtn" ).Visible = msoFalse
78      End If
79
80      End Sub
```

## A

Application, [26](#)

## C

Cal, [26](#)  
CalendarHide, [26](#)  
CalendarShow, [26](#)  
Cells, [26](#)  
CheckForSheet, [26](#)  
ClearContents, [27](#)  
Column, [26](#), [27](#)  
CountLarge, [26](#)

## E

Empty, [26](#), [27](#)  
EnableEvents, [26](#)  
Explicit, [26](#)

## I

Intersect, [26](#), [27](#)

## M

msoFalse, [27](#)

## P

Payroll\_FilterEmployees, [26](#)  
Payroll\_Load\_Review, [26](#)  
Payroll\_Load\_Summary, [26](#)  
Payroll\_LoadEmpReview, [27](#)  
Payroll\_ShowEditEmployeeIcon, [27](#)  
Payroll\_ShowEditTimeClockIcon, [27](#)

## R

Range, [26](#), [27](#)  
Row, [26](#), [27](#)

## S

Shape, [26](#)  
Shapes, [26](#), [27](#)  
Sheet18, [26](#)  
Sheet4, [26](#)

## T

Target, [26](#), [27](#)

## U

Undo, [26](#)  
UpdateLocal, [26](#)

## V

Value, [26](#), [27](#)  
Visible, [27](#)

## W

Worksheet\_Activate, [26](#)  
Worksheet\_Change, [26](#)  
Worksheet\_SelectionChange, [26](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A3:I999999")) Is Nothing Then
9         If Sheet4.Range("B1").Value = False And Target.Count < 2 And Sheet4.Range("J17").
            Value = "On" And Sheet4.Range("ApplicationFolder").Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject")
17            UserName = Sheet14.Range("AppUserName").Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow").Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder").Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30 End Sub
```

## A

Address, [29](#)

## C

Close, [29](#)

Count, [29](#)

createFolder, [29](#)

CreateObject, [29](#)

CreateTextFile, [29](#)

## D

Dir, [29](#)

## E

Empty, [29](#)

Explicit, [29](#)

## F

FileName, [29](#)

fso, [29](#)

## I

Intersect, [29](#)

## L

LastUser, [29](#)

## N

Name, [29](#)

## O

oFile, [29](#)

## R

Range, [29](#)

## S

Sheet14, [29](#)

Sheet4, [29](#)

## T

Target, [29](#)

## U

UpdateLocal, [29](#)

UserName, [29](#)

UserRow, [29](#)

## V

Value, [29](#)

vbDirectory, [29](#)

## W

Worksheet, [29](#)

Worksheet\_Activate, [29](#)

Worksheet\_Change, [29](#)

WriteLine, [29](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A3:F999999")) Is Nothing Then
9         If Sheet4.Range("B1").Value = False And Target.Count < 2 And Sheet4.Range("J17").
            Value = "On" And Sheet4.Range("ApplicationFolder").Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject")
17            UserName = Sheet14.Range("AppUserName").Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow").Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder").Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30 End Sub
```

## A

Address, [31](#)

## C

Close, [31](#)

Count, [31](#)

createFolder, [31](#)

CreateObject, [31](#)

CreateTextFile, [31](#)

## D

Dir, [31](#)

## E

Empty, [31](#)

Explicit, [31](#)

## F

FileName, [31](#)

fso, [31](#)

## I

Intersect, [31](#)

## L

LastUser, [31](#)

## N

Name, [31](#)

## O

oFile, [31](#)

## R

Range, [31](#)

## S

Sheet14, [31](#)

Sheet4, [31](#)

## T

Target, [31](#)

## U

UpdateLocal, [31](#)

UserName, [31](#)

UserRow, [31](#)

## V

Value, [31](#)

vbDirectory, [31](#)

## W

Worksheet, [31](#)

Worksheet\_Activate, [31](#)

Worksheet\_Change, [31](#)

WriteLine, [31](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A3:J999999")) Is Nothing Then
9         If Sheet4.Range("B1").Value = False And Target.Count < 2 And Sheet4.Range("J17").
            Value = "On" And Sheet4.Range("ApplicationFolder").Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject")
17            UserName = Sheet14.Range("AppUserName").Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow").Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder").Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder").Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30 End Sub
31
```

## A

Address, [33](#)

## C

Close, [33](#)

Count, [33](#)

createFolder, [33](#)

CreateObject, [33](#)

CreateTextFile, [33](#)

## D

Dir, [33](#)

## E

Empty, [33](#)

Explicit, [33](#)

## F

FileName, [33](#)

fso, [33](#)

## I

Intersect, [33](#)

## L

LastUser, [33](#)

## N

Name, [33](#)

## O

oFile, [33](#)

## R

Range, [33](#)

## S

Sheet14, [33](#)

Sheet4, [33](#)

## T

Target, [33](#)

## U

UpdateLocal, [33](#)

UserName, [33](#)

UserRow, [33](#)

## V

Value, [33](#)

vbDirectory, [33](#)

## W

Worksheet, [33](#)

Worksheet\_Activate, [33](#)

Worksheet\_Change, [33](#)

WriteLine, [33](#)



```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A4:AG999999" )) Is Nothing Then
9         If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
            Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject" )
17            UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30 End Sub
31
```

## A

Address, [35](#)

## C

Close, [35](#)

Count, [35](#)

createFolder, [35](#)

CreateObject, [35](#)

CreateTextFile, [35](#)

## D

Dir, [35](#)

## E

Empty, [35](#)

Explicit, [35](#)

## F

FileName, [35](#)

fso, [35](#)

## I

Intersect, [35](#)

## L

LastUser, [35](#)

## N

Name, [35](#)

## O

oFile, [35](#)

## R

Range, [35](#)

## S

Sheet14, [35](#)

Sheet4, [35](#)

## T

Target, [35](#)

## U

UpdateLocal, [35](#)

UserName, [35](#)

UserRow, [35](#)

## V

Value, [35](#)

vbDirectory, [35](#)

## W

Worksheet, [35](#)

Worksheet\_Activate, [35](#)

Worksheet\_Change, [35](#)

WriteLine, [35](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A3:E999999" )) Is Nothing Then
9         If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
            Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject" )
17            UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30 End Sub
```

## A

Address, [37](#)

## C

Close, [37](#)

Count, [37](#)

createFolder, [37](#)

CreateObject, [37](#)

CreateTextFile, [37](#)

## D

Dir, [37](#)

## E

Empty, [37](#)

Explicit, [37](#)

## F

FileName, [37](#)

fso, [37](#)

## I

Intersect, [37](#)

## L

LastUser, [37](#)

## N

Name, [37](#)

## O

oFile, [37](#)

## R

Range, [37](#)

## S

Sheet14, [37](#)

Sheet4, [37](#)

## T

Target, [37](#)

## U

UpdateLocal, [37](#)

UserName, [37](#)

UserRow, [37](#)

## V

Value, [37](#)

vbDirectory, [37](#)

## W

Worksheet, [37](#)

Worksheet\_Activate, [37](#)

Worksheet\_Change, [37](#)

WriteLine, [37](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("D3:E3" )) Is Nothing And Range("B3" ).Value = False Then
9         Attach_Filter
10
11     If Not Intersect(Target, Range("D5:K999999" )) Is Nothing Then 'We dont want to run when
12         user makes filter cell changes, that is a per-user filter
13     If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
14         Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check for
15         Sync Off & Application Sharing On and Shared Folder
16         Dim UserRow As Long
17         Dim LastUser As Long
18         Dim UserName As String
19         Dim FileName As String
20         Dim fso As Object
21         Dim oFile As Object
22         Set fso = CreateObject("Scripting.FileSystemObject" )
23         UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
24         LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
25         For UserRow = 308 To LastUser
26             If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
27                 UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
28                 "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
29             If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
30                 UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
31                 "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
32             FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
33                 UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
34             Set oFile = fso.CreateTextFile(FileName)
35             oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
36                 Value
37             oFile.Close
38         Next UserRow
39         Set fso = Nothing
40         Set oFile = Nothing
41     End If
42 End If
43 End Sub
44
45
```

## A

Address, [39](#)  
Attach\_Filter, [39](#)

## C

Close, [39](#)  
Count, [39](#)  
createFolder, [39](#)  
CreateObject, [39](#)  
CreateTextFile, [39](#)

## D

Dir, [39](#)

## E

Empty, [39](#)  
Explicit, [39](#)

## F

FileName, [39](#)  
fso, [39](#)

## I

Intersect, [39](#)

## L

LastUser, [39](#)

## N

Name, [39](#)

## O

oFile, [39](#)

## R

Range, [39](#)

## S

Sheet14, [39](#)  
Sheet4, [39](#)

## T

Target, [39](#)

## U

UpdateLocal, [39](#)  
UserName, [39](#)  
UserRow, [39](#)

## V

Value, [39](#)  
vbDirectory, [39](#)

## W

Worksheet, [39](#)  
Worksheet\_Activate, [39](#)  
Worksheet\_Change, [39](#)  
WriteLine, [39](#)

```

1 Option Explicit
2 Dim TimeFormat As Long
3
4 Private Sub Worksheet_Activate()
5     ActiveWindow.ScrollRow = 1
6     UpdateLocal
7 End Sub
8
9 Private Sub Worksheet_Change(ByVal Target As Range)
10    On Error Resume Next
11
12    'On Change Of Custom Field Time
13    If Not Intersect(Target, Range("G108:G111" )) Is Nothing Then
14        Dim FieldTypeNumber As Long
15        FieldTypeNumber = Application.WorksheetFunction.Match(Target.Value, Sheet4.Range(
16            "FieldTypes" ), 0)
17        If FieldTypeNumber <> 0 Then
18            If FieldTypeNumber <> 5 Then
19                Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation.Delete 'Delete
20                Any possible Data validation
21                Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).NumberFormat = Sheet4.Range(
22                    "B" & 119 + FieldTypeNumber).Value
23            Else:
24                With Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation
25                    .Delete
26                    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween,
27                    Formula1:="=CustomList" & Target.Row - 107
28                    .IgnoreBlank = True
29                    .InCellDropdown = True
30                    .ShowInput = True
31                    .ShowError = True
32                End With
33            End If
34        End If
35    End If
36
37    'On Change Of Existing (not New) Notifications, update table below, and not on
38    Notification load
39    If Not Intersect(Target, Range("E507:E509,D511,G507,G508,I508" )) Is Nothing And Range(
40        "B506" ).Value = False And Range("B508" ).Value = False Then
41        Dim NotifRow As Long
42        Dim NotifCol As Long
43        NotifRow = Range("B507" ).Value 'Get Table Row
44        NotifCol = Cells(Target.Row, Target.Column + 20).Value 'Get Table Column
45        Cells(NotifRow, NotifCol).Value = Target.Value 'Add Change to Table Below
46    End If
47
48    'On Change of Global Time Format, run Time format macro
49    If Not Intersect(Target, Range("J12" )) Is Nothing Then Admin_SetTotalFormat
50
51    If Not Intersect(Target, Range("C5:M504,C521:M541" )) Is Nothing Then
52        If Intersect(Target, Range("J16,B1,J18,K18" )) Is Nothing Then 'Run on any change
53            except J16 which is the local folder location, Sync Frequency Values which are
54            user-based
55            If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
56                Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check
57                for Sync Off & Application Sharing On and Shared Folder
58                Dim UserRow As Long
59                Dim LastUser As Long
60                Dim UserName As String
61                Dim FileName As String

```

1 2 3 4

```

52   1 2 3 4 Dim fso As Object
53   Dim oFile As Object
54   Set fso = CreateObject("Scripting.FileSystemObject" )
55   UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
56   LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
57   For UserRow = 308 To LastUser
58       If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
59           & UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
60           "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
61           FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E"
62           & UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
63           Set oFile = fso.CreateTextFile(FileName)
64           oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
65           Value
66           oFile.Close
67       Next UserRow
68       Set fso = Nothing
69       Set oFile = Nothing
70   End If
71   End If
72   End If
73   If Not Intersect(Target, Range("J17,J18,K18" )) Is Nothing Then SetSyncFrequency 'Run
74   Sync Frequency Macro on change
75   End Sub
76
77 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
78     If Target.CountLarge > 1 Then Exit Sub
79     If Not Intersect(Target, Range("D4:K4" )) Is Nothing Then
80         Range("B3" ).Value = Target.Column
81         Range("F2" ).Select
82         Admin_SwitchHorizontalTabs
83     End If
84
85     'Set Checkbox for User To Send Emails or Display Notification Pop-ups (Make changes only
86     on existing users, not blank rows
87     If Not Intersect(Target, Range("I308:I404" )) Is Nothing And Range("E" & Target.Row).
88     Value <> Empty Then
89         Range("I308:I404" ).ClearContents 'Clear any Previous checkboxes
90         Target.Value = "ü"
91     End If
92
93     If Not Intersect(Target, Range("D407:L518" )) Is Nothing Then
94         'On Selection of Notification Fields, set Tab order, otherwise cancel tab order
95         If Target.Row > 506 And Target.Row < 518 Then SetOnkey (True)
96
97         'On Selection of Notification Fields, set Tab order, otherwise cancel tab order
98         If Target.Row > 407 And Target.Row < 428 Then SetOnkey (True) '
99     Else: SetOnkey (False)
100    End If
101
102    If Not Intersect(Target, Range("E308:I404" )) Is Nothing And Range("E" & Target.Row).
103    Value <> Empty Then
104        Range("B308" ).Value = Target.Row
105    Else: 'Range("B308").ClearContents
106    End If
107
108    'Selection of Notifications to Load in cells above
109    If Not Intersect(Target, Range("D521:L541" )) Is Nothing And Range("D" & Target.Row).
110    Value <> Empty Then

```



```
104   1 2   Range("B507" ).Value = Target.Row
105       Notif_Load
106       End If
107   End Sub
```

**A**

ActiveWindow, 41  
Add, 41  
Address, 42  
Admin\_SetTotalFormat, 41  
Admin\_SwitchHorizontalTabs, 42  
AlertStyle, 41  
Application, 41

**C**

Cells, 41  
ClearContents, 42  
Close, 42  
Column, 41, 42  
Count, 41  
CountLarge, 42  
createFolder, 42  
CreateObject, 42  
CreateTextFile, 42

**D**

Delete, 41  
Dir, 42

**E**

Empty, 41, 42  
Explicit, 41

**F**

FieldTypeNumber, 41  
FileName, 41, 42  
Formula1, 41  
fso, 42

**I**

IgnoreBlank, 41  
InCellDropdown, 41  
Intersect, 41, 42

**L**

LastUser, 41, 42

**M**

Match, 41

**N**

Name, 42  
Notif\_Load, 43  
NotifCol, 41  
NotifRow, 41  
NumberFormat, 41

**O**

oFile, 42  
Operator, 41

**R**

Range, 41-43  
Row, 41-43

**S**

ScrollRow, 41  
SetOnkey, 42  
SetSyncFrequency, 42  
Sheet1, 41  
Sheet14, 42  
Sheet4, 41, 42  
ShowError, 41  
ShowInput, 41

**T**

Target, 41-43  
TimeFormat, 41

**U**

UpdateLocal, 41  
UserName, 41, 42  
UserRow, 41, 42

**V**

Validation, 41  
Value, 41-43  
vbDirectory, 42

**W**

Worksheet, 42  
Worksheet\_Activate, 41  
Worksheet\_Change, 41  
Worksheet\_SelectionChange, 42  
WorksheetFunction, 41  
WriteLine, 42

**X**

xlBetween, 41  
xlValidAlertStop, 41  
xlValidateList, 41

Option Explicit

```
Private Sub Worksheet_Activate()
    UpdateLocal
End Sub
```

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

```
    Dim MapRng As Range
```

```
    Dim FoundMapRng As Range
```

```
    Dim CellAdd As String
```

```
    Dim SelRow As Long
```

```
    If Not Intersect(Target, Range("L13" )) Is Nothing Then
```

```
        If Target.Value = "Yes" Then Range("14:21" ).EntireRow.Hidden = False Else: Range(
            "14:21" ).EntireRow.Hidden = True
```

```
    End If
```

```
'On Any Existing Event Change Update Event List Table AND Selected Item in List below
Events, But not on Event Load
```

```
If Not Intersect(Target, Range("D4:N21" )) Is Nothing And Range("B2" ).Value = False And
Range("B1" ).Value = False And IsNumeric(Cells(Target.Row, Target.Column + 23).Value) =
True And Cells(Target.Row, Target.Column + 23).Value <> Empty Then
```

```
    Dim EventRow As Long
```

```
    If Not Intersect(Target, Range("F13:G13" )) Is Nothing Then Sheet7.Range("AA" & Range(
        "B3" ).Value).Value = Range("J10" ).Value + Range("L10" ).Value - Range("B11" ).Value
```

```
'Update Reminder On Date & Time
```

```
EventRow = Range("B3" ).Value 'Event List Row
```

```
If Target.Column = 4 Then 'Reminders & Recurring Options
```

```
    If InStr(Target.Value, "p" ) > 0 Then
```

```
        Sheet7.Cells(EventRow, Cells(Target.Row, Target.Column + 23).Value).Value =
            "Yes"
```

```
        Sheet7.Range("AA" & EventRow).Value = Range("J10" ).Value + Range("L10" ).Value
            - Range("B11" ).Value 'Set Reminder On Date & Time
```

```
    Else: Sheet7.Cells(EventRow, Cells(Target.Row, Target.Column + 23).Value).Value =
        "No"
```

```
    End If
```

```
Else: 'All Other fields
```

```
    Sheet7.Cells(EventRow, Cells(Target.Row, Target.Column + 23).Value).Value = Target.
        Value
```

```
End If
```

```
Sheet7.Range(EventRow & ":" & EventRow).WrapText = False
```

```
'Update Table Below
```

```
CellAdd = Target.Address
```

```
SelRow = Range("B10" ).Value
```

```
Set MapRng = Sheet5.Range("EventDataMap" )
```

```
Set FoundMapRng = MapRng.Find(CellAdd, , xlValues, xlWhole)
```

```
If Not FoundMapRng Is Nothing Then Cells(SelRow, FoundMapRng.Column).Value = Sheet7.
    Cells(Range("B3" ).Value, Cells(Target.Row, Target.Column + 23).Value).Value
```

```
End If
```

```
'On change of employee, update Employee ID and Email Address, but not on Event Load
```

```
If Not Intersect(Target, Range("K4" )) Is Nothing And Range("B5" ).Value <> Empty And
Range("B1" ).Value = False Then
```

```
    Range("F15" ).Value = Sheet2.Range("M" & Range("B5" ).Value).Value 'Add Employee
    Email
```

```
    Range("B6" ).Value = Sheet2.Range("A" & Range("B5" ).Value).Value
```

```
    If Range("B3" ).Value <> Empty Then Sheet7.Range("E" & Range("B3" ).Value).Value =
        Range("B6" ).Value 'Update Employee ID on Employee Change
```

```
End If
```

```
'Add in Default Notifications but not on Event Load and only for New Events
```

```
If Not Intersect(Target, Range("H4" )) Is Nothing And Range("B1" ).Value = False And Range
```

```

1      ("B2" ).Value = True Then Event_AddNotifDefaults
48
49      'Run Table Filter on change but not when text contains the word "Enter"
50      On Error Resume Next
51      If Not Intersect(Target, Range("D26,E26,G26" )) Is Nothing Then
52          If InStr(Target.Value, "Enter" ) = 0 Then Events_FilterTable
53      End If
54
55      On Error GoTo 0
56      'On Any Change of Event Type, Show All Day Button for Training and Leave Type Events
57      If Not Intersect(Target, Range("H4" )) Is Nothing Then
58
59
60          If Range("H4" ).Value = "Leave" Or Range("H4" ).Value = "Training" Then Shapes(
        "AllDayBtn" ).Visible = msoCTrue Else: Shapes("AllDayBtn" ).Visible = msoFalse
61      End If
62
63      End Sub
64
65      Private Sub Worksheet_Deactivate()
66          Dim TP As Shape
67          On Error Resume Next
68          Set TP = Shapes("TimePickGrp" )
69          On Error GoTo 0
70          If TP Is Nothing Then 'Shape Deleted
71              Application.EnableEvents = False
72              On Error Resume Next
73              Application.Undo
74              Application.EnableEvents = True
75          End If
76      End Sub
77
78      Private Sub Worksheet_SelectionChange(ByVal Target As Range)
79          If Target.CountLarge > 1 Then Exit Sub
80          'For Pop-up Calendar
81          Dim Cal As Shape 'Calendar
82          On Error Resume Next
83          Set Cal = Shapes("Calendar" )
84          On Error GoTo 0
85          If Cal Is Nothing Then 'Shape Deleted
86              Application.EnableEvents = False
87              On Error Resume Next
88              Application.Undo
89              On Error GoTo 0
90              Application.EnableEvents = True
91          End If
92          If Not Intersect(Target, Range("J10,J11" )) Is Nothing Then 'Change this to any cell (or
        cells) you would like to have the Pop-Up Calendar Appear
93              CheckForSheet
94              CalendarShow
95          Else:
96              CheckForSheet
97              CalendarHide
98          End If
99
100         'For Pop-up Date Picker
101         Dim TP As Shape 'Time Picker
102         On Error Resume Next
103         Set TP = Shapes("TimePickGrp" )
104         On Error GoTo 0
105         If TP Is Nothing Then 'Shape Deleted

```

```
1 2
106 Application.EnableEvents = False
107 On Error Resume Next
108 Application.Undo
109 On Error GoTo 0
110 Application.EnableEvents = True
111 End If
112 If Not Intersect(Target, Range("L10,L11" )) Is Nothing Then
113     'If Sheet is Protected, add Unprotect Code here such as ActiveSheet.Unprotect
114     "Password"
115     TPSHOW
116 Else:
117     Shapes("TimePickGrp" ).Visible = msoFalse
118 End If
119 If Not Intersect(Target, Range("E4:N21" )) Is Nothing Then
120     SetOnkey (True)
121 Else:
122     SetOnkey (False)
123 End If
124
125 'On Table Selection
126 If Not Intersect(Target, Range("D27:N9999" )) Is Nothing And Range("D" & Target.Row).
127 Value <> Empty Then
128     Range("B10" ).Value = Target.Row 'Add in Selected Row
129     Range("B4" ).Value = Range("O" & Target.Row).Value 'Add In Event ID
130     Event_Load
131 End If
132
133 'On Recurring Selection
134 If Not Intersect(Target, Range("D10" )) Is Nothing Then
135     With ActiveCell
136         If .Value = "p    Recurring" Then .Value = "o    Recurring" Else: .Value = "p
137             Recurring"
138         .Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
139         .Characters(Start:=2, Length:=16).Font.Name = "Calibri"
140     End With
141     Range("F11" ).Select
142 End If
143
144 'On Reminders selection
145 If Not Intersect(Target, Range("D13" )) Is Nothing Then
146     With ActiveCell
147         If .Value = "p    Reminders" Then .Value = "o    Reminders" Else: .Value = "p
148             Reminders"
149         .Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
150         .Characters(Start:=2, Length:=15).Font.Name = "Calibri"
151         If .Value = "p    Reminders" Then
152             Sheet5.Shapes("EventRemindIcons" ).Visible = msoCTrue
153         Else:
154             Sheet5.Shapes("EventRemindIcons" ).Visible = msoFalse
155             Range("L13" ).Value = "No"
156         End If
157     End With
158     Range("I13" ).Select
159 End If
End Sub
```

**A**

ActiveCell, 47  
Address, 45  
Application, 46, 47

**C**

Cal, 46  
CalendarHide, 46  
CalendarShow, 46  
CellAdd, 45  
Cells, 45  
Characters, 47  
CheckForSheet, 46  
Column, 45  
CountLarge, 46

**E**

Empty, 45, 47  
EnableEvents, 46, 47  
EntireRow, 45  
Event\_AddNotifDefaults, 46  
Event\_Load, 47  
EventRow, 45  
Events\_FilterTable, 46  
Explicit, 45

**F**

Find, 45  
Font, 47  
FoundMapRng, 45

**H**

Hidden, 45

**I**

InStr, 45, 46  
Intersect, 45-47  
IsNumeric, 45

**L**

Length, 47

**M**

MapRng, 45  
msoCTrue, 46, 47  
msoFalse, 46, 47

**N**

Name, 47

**R**

Range, 45-47  
Row, 45, 47

**S**

SelRow, 45  
SetOnkey, 47  
Shape, 46  
Shapes, 46, 47  
Sheet2, 45  
Sheet5, 45, 47  
Sheet7, 45  
Start, 47

**T**

Target, 45-47  
TP, 46  
TPShow, 47

**U**

Undo, 46, 47  
UpdateLocal, 45

**V**

Value, 45-47  
Visible, 46, 47

**W**

Worksheet\_Activate, 45  
Worksheet\_Change, 45  
Worksheet\_Deactivate, 46  
Worksheet\_SelectionChange, 46  
WrapText, 45

**X**

xlValues, 45  
xlWhole, 45

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A4:AG999999" )) Is Nothing Then
9         If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
            Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject" )
17            UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30 End Sub
31
32 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
33     If Not Intersect(Target, Range("A4:AG999" )) Is Nothing And Range("A" & Target.Row).Value
        <> Empty Then
34         Range("BA1" ).Value = Target.Row
35     Else: Range("BA1" ).ClearContents
36     End If
37 End Sub
```

## A

Address, [49](#)

## C

ClearContents, [49](#)

Close, [49](#)

Count, [49](#)

createFolder, [49](#)

CreateObject, [49](#)

CreateTextFile, [49](#)

## D

Dir, [49](#)

## E

Empty, [49](#)

Explicit, [49](#)

## F

FileName, [49](#)

fso, [49](#)

## I

Intersect, [49](#)

## L

LastUser, [49](#)

## N

Name, [49](#)

## O

oFile, [49](#)

## R

Range, [49](#)

Row, [49](#)

## S

Sheet14, [49](#)

Sheet4, [49](#)

## T

Target, [49](#)

## U

UpdateLocal, [49](#)

UserName, [49](#)

UserRow, [49](#)

## V

Value, [49](#)

vbDirectory, [49](#)

## W

Worksheet, [49](#)

Worksheet\_Activate, [49](#)

Worksheet\_Change, [49](#)

Worksheet\_SelectionChange, [49](#)

WriteLine, [49](#)



```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A5:AB999999" )) Is Nothing Then
9         If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
            Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check for
            Sync Off & Application Sharing On and Shared Folder
10            Dim UserRow As Long
11            Dim LastUser As Long
12            Dim UserName As String
13            Dim FileName As String
14            Dim fso As Object
15            Dim oFile As Object
16            Set fso = CreateObject("Scripting.FileSystemObject" )
17            UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
18            LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
19            For UserRow = 308 To LastUser
20                If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
                    "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
21                FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
22                Set oFile = fso.CreateTextFile(FileName)
23                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
                    Value
24                oFile.Close
25            Next UserRow
26            Set fso = Nothing
27            Set oFile = Nothing
28        End If
29    End If
30 End Sub
31
32
```

## A

Address, [51](#)

## C

Close, [51](#)

Count, [51](#)

createFolder, [51](#)

CreateObject, [51](#)

CreateTextFile, [51](#)

## D

Dir, [51](#)

## E

Empty, [51](#)

Explicit, [51](#)

## F

FileName, [51](#)

fso, [51](#)

## I

Intersect, [51](#)

## L

LastUser, [51](#)

## N

Name, [51](#)

## O

oFile, [51](#)

## R

Range, [51](#)

## S

Sheet14, [51](#)

Sheet4, [51](#)

## T

Target, [51](#)

## U

UpdateLocal, [51](#)

UserName, [51](#)

UserRow, [51](#)

## V

Value, [51](#)

vbDirectory, [51](#)

## W

Worksheet, [51](#)

Worksheet\_Activate, [51](#)

Worksheet\_Change, [51](#)

WriteLine, [51](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     Sched_Refresh
5     UpdateLocal
6 End Sub
7
8 Private Sub Worksheet_Change(ByVal Target As Range)
9     'Clear Comments on Year Change
10    If Not Intersect(Target, Range("E2" )) Is Nothing Then Range("F3:DG309" ).ClearComments
11    'Clear All Comments
12    If Not Intersect(Target, Range("D3" )) Is Nothing Then
13        If Range("B3" ).Value <> Empty Then Sched_ScrollToRow Else: ActiveWindow.ScrollRow = 7
14    End If
15
16    'Scheduling Filters (Type/Position)
17    If Not Intersect(Target, Range("DD3,T3" )) Is Nothing Then
18        If Target.Value <> "[All Types/Positions]" Then
19            If ActiveSheet.AutoFilterMode Then AutoFilter.ShowAllData
20            Range("$C$6:$FP$314" ).AutoFilter
21            Range("$C$6:$FP$314" ).AutoFilter Field:=170, Criteria1:=Target.Value
22        Else: If ActiveSheet.AutoFilterMode Then AutoFilter.ShowAllData '=[All
23            Types/Positions]
24        End If
25    End If
26
27    'Scheduling Filters (Event Types)
28    If Not Intersect(Target, Range("AC3,DF3" )) Is Nothing Then
29        Dim AddCol As Long
30        If Target.Address = "$AC$3" Then AddCol = 1
31        If Target.Value <> "[All Event Types]" Then
32            If ActiveSheet.AutoFilterMode Then AutoFilter.ShowAllData
33            Range("$C$6:$FR$314" ).AutoFilter
34            Range("$C$6:$FR$314" ).AutoFilter Field:=171 + AddCol, Criteria1:="True"
35        Else: If ActiveSheet.AutoFilterMode Then AutoFilter.ShowAllData
36        End If
37    End If
38 End Sub
39
40 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
41     If Target.Count > 5 Then Exit Sub
42     If Not Intersect(Target, Range("F2" )) Is Nothing Then Sched_WeekView
43     If Not Intersect(Target, Range("DB2" )) Is Nothing Then Sched_MonthView
44     If Not Intersect(Target, Range("F8:DG309" )) Is Nothing Then
45         If Cells(Target.Row, Target.Column + 60).Value <> Empty Then Sched_AddComment_Event
46         If Cells(5, Target.Column).Value = "H" Then Sched_AddComment_Holiday
47         Application.DisplayCommentIndicator = xlCommentIndicatorOnly
48     End If
49 End Sub
```

## A

ActiveSheet, [53](#)  
ActiveWindow, [53](#)  
AddCol, [53](#)  
Address, [53](#)  
Application, [53](#)  
AutoFilter, [53](#)  
AutoFilterMode, [53](#)

## C

Cells, [53](#)  
ClearComments, [53](#)  
Column, [53](#)  
Count, [53](#)  
Criteria1, [53](#)

## D

DisplayCommentIndicator, [53](#)

## E

Empty, [53](#)  
Explicit, [53](#)

## F

Field, [53](#)

## I

Intersect, [53](#)

## R

Range, [53](#)  
Row, [53](#)

## S

Sched\_AddComment\_Event, [53](#)  
Sched\_AddComment\_Holiday, [53](#)  
Sched\_MonthView, [53](#)  
Sched\_Refresh, [53](#)  
Sched\_ScrollToRow, [53](#)  
Sched\_WeekView, [53](#)  
ScrollRow, [53](#)  
ShowAllData, [53](#)

## T

Target, [53](#)

## U

UpdateLocal, [53](#)

## V

Value, [53](#)

## W

Worksheet\_Activate, [53](#)  
Worksheet\_Change, [53](#)  
Worksheet\_SelectionChange, [53](#)

## X

xlCommentIndicatorOnly, [53](#)

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     UpdateLocal
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     If Not Intersect(Target, Range("A5:J999999" )) Is Nothing Then
9         If Sheet4.Range("B1" ).Value = False And Target.Count < 2 And Sheet4.Range("J17" ).
10            Value = "On" And Sheet4.Range("ApplicationFolder" ).Value <> Empty Then 'Check for
11            Sync Off & Application Sharing On and Shared Folder
12            Dim UserRow As Long
13            Dim LastUser As Long
14            Dim UserName As String
15            Dim FileName As String
16            Dim fso As Object
17            Dim oFile As Object
18            Set fso = CreateObject("Scripting.FileSystemObject" )
19            UserName = Sheet14.Range("AppUserName" ).Value 'Current User Name
20            LastUser = Sheet4.Range("LastUserRow" ).Value 'Last User Row
21            For UserRow = 308 To LastUser
22                If Dir(Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
23                    UserRow).Value, vbDirectory) = "" Then fso.createFolder (Sheet4.Range(
24                    "ApplicationFolder" ).Value & "\" & Sheet4.Range("E" & UserRow).Value)
25                FileName = Sheet4.Range("ApplicationFolder" ).Value & "\" & Sheet4.Range("E" &
26                    UserRow).Value & "\" & Target.Worksheet.Name & Target.Address & ".txt"
27                Set oFile = fso.CreateTextFile(FileName)
28                oFile.WriteLine Target.Worksheet.Name & "," & Target.Address & ":" & Target.
29                    Value
30                oFile.Close
31            Next UserRow
32            Set fso = Nothing
33            Set oFile = Nothing
34        End If
35    End If
36 End Sub
37
```

## A

Address, [55](#)

## C

Close, [55](#)

Count, [55](#)

createFolder, [55](#)

CreateObject, [55](#)

CreateTextFile, [55](#)

## D

Dir, [55](#)

## E

Empty, [55](#)

Explicit, [55](#)

## F

FileName, [55](#)

fso, [55](#)

## I

Intersect, [55](#)

## L

LastUser, [55](#)

## N

Name, [55](#)

## O

oFile, [55](#)

## R

Range, [55](#)

## S

Sheet14, [55](#)

Sheet4, [55](#)

## T

Target, [55](#)

## U

UpdateLocal, [55](#)

UserName, [55](#)

UserRow, [55](#)

## V

Value, [55](#)

vbDirectory, [55](#)

## W

Worksheet, [55](#)

Worksheet\_Activate, [55](#)

Worksheet\_Change, [55](#)

WriteLine, [55](#)

```
1 Option Explicit
2
3 Private Sub Workbook_Activate()
4     If ActiveSheet.CodeName = "Sheet1" Then SetOnkey (True) Else: SetOnkey (False)
5
6     'Uncomment The Macro below to automatically update remote time clock entries on Workbook
7     open
8     'TimeClock_SyncRemoteEntries
9
10    'Uncomment the StartWorkbook macro below to run the User Login Screen & Pop_Up
11    'StartWorkbook
12    'UpdateLocal 'Update Local Copy on workbook open when in Sharing mode
13 End Sub
14
15 Private Sub Workbook_BeforeClose(Cancel As Boolean)
16     On Error Resume Next
17     Application.CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
18     Application.CommandBars("Cell" ).Controls("Go To Event" ).Delete
19     Application.CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
20     On Error GoTo 0
21 End Sub
22
23 Private Sub Workbook_Deactivate()
24     SetOnkey (False)
25     On Error Resume Next
26     Application.CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
27     Application.CommandBars("Cell" ).Controls("Go To Event" ).Delete
28     Application.CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
29     On Error GoTo 0
30 End Sub
31
32 Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel
33     As Boolean)
34     On Error Resume Next
35     Application.CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
36     Application.CommandBars("Cell" ).Controls("Go To Event" ).Delete
37     Application.CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
38     On Error GoTo 0
39     If ActiveSheet.CodeName = "Sheet8" Then RightClick_CallEvents
40 End Sub
```

## A

ActiveSheet, [57](#)  
Application, [57](#)

## C

Cancel, [57](#)  
CodeName, [57](#)  
CommandBars, [57](#)  
Controls, [57](#)

## D

Delete, [57](#)

## E

Explicit, [57](#)

## R

Range, [57](#)  
RightClick\_CallEvents, [57](#)

## S

SetOnkey, [57](#)  
Sh, [57](#)

## T

Target, [57](#)

## W

Workbook\_Activate, [57](#)  
Workbook\_BeforeClose, [57](#)  
Workbook\_Deactivate, [57](#)  
Workbook\_SheetBeforeRightClick, [57](#)



```
1  
2  
3 Option Explicit  
4  
5 Private Sub CancelBtn_Click()  
6     CreatePayPeriod.Hide  
7 End Sub  
8  
9 Private Sub ClsBtn_Click()  
10    CreatePayPeriod.Hide  
11 End Sub  
12  
13 Private Sub CreateBtn_Click()  
14    CreatePayPeriods  
15 End Sub  
16
```

## C

CancelBtn\_Click, [59](#)  
ClsBtn\_Click, [59](#)  
CreateBtn\_Click, [59](#)  
CreatePayPeriod, [59](#)  
CreatePayPeriods, [59](#)

## E

Explicit, [59](#)

## H

Hide, [59](#)

```
1 Option Explicit
2
3 Private Sub BreakIn_AfterUpdate()
4     EditTimeClock_Calculate
5 End Sub
6
7 Private Sub BreakInNow_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As
8 Single, ByVal Y As Single)
9     BreakIn.Value = Format(Now, "h:mm AM/PM" )
10    EditTimeClock_Calculate
11 End Sub
12
13 Private Sub BreakOut_AfterUpdate()
14     EditTimeClock_Calculate
15 End Sub
16
17 Private Sub BreakOutNow_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As
18 Single, ByVal Y As Single)
19     BreakOut.Value = Format(Now, "h:mm AM/PM" )
20    EditTimeClock_Calculate
21 End Sub
22
23
24
25 Private Sub CancelBtn_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As
26 Single, ByVal Y As Single)
27     EditTimeClock.Hide
28 End Sub
29
30 Private Sub ClockIn_AfterUpdate()
31     EditTimeClock_Calculate
32 End Sub
33
34
35
36
37 Private Sub ClockInNow_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As
38 Single, ByVal Y As Single)
39     ClockIn.Value = Format(Now, "h:mm AM/PM" )
40    EditTimeClock_Calculate
41 End Sub
42
43 Private Sub ClockOut_AfterUpdate()
44     EditTimeClock_Calculate
45 End Sub
46
47 Private Sub ClockOutNow_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As
48 Single, ByVal Y As Single)
49     ClockOut.Value = Format(Now, "h:mm AM/PM" )
50    EditTimeClock_Calculate
51 End Sub
52
53
54 Private Sub OkBtn_Click()
55     EditTimeClock_Save
56 End Sub
```

```
57  
58 Private Sub UserForm_Activate()  
59     ClockIn.Text = Format(ClockIn.Text, "h:mm AM/PM" )  
60     BreakOut.Text = Format(BreakOut.Text, "h:mm AM/PM" )  
61     BreakIn.Text = Format(BreakIn.Text, "h:mm AM/PM" )  
62     ClockOut.Text = Format(ClockOut.Text, "h:mm AM/PM" )  
63     EmplName.Text = Sheet8.Range("A55" ).Value 'Employee Name  
64 End Sub  
65  
66  
67
```

## B

BreakIn, [61](#), [62](#)  
BreakIn\_AfterUpdate, [61](#)  
BreakInNow\_MouseDown, [61](#)  
BreakOut, [61](#), [62](#)  
BreakOut\_AfterUpdate, [61](#)  
BreakOutNow\_MouseDown, [61](#)  
Button, [61](#)

## C

CancelBtn\_MouseDown, [61](#)  
ClockIn, [61](#), [62](#)  
ClockIn\_AfterUpdate, [61](#)  
ClockInNow\_MouseDown, [61](#)  
ClockOut, [61](#), [62](#)  
ClockOut\_AfterUpdate, [61](#)  
ClockOutNow\_MouseDown, [61](#)

## E

EditTimeClock, [61](#)  
EditTimeClock\_Calculate, [61](#)  
EditTimeClock\_Save, [61](#)  
EmplName, [62](#)  
Explicit, [61](#)

## F

Format, [61](#), [62](#)

## H

Hide, [61](#)

## N

Now, [61](#)

## O

OkBtn\_Click, [61](#)

## R

Range, [62](#)

## S

Sheet8, [62](#)  
Shift, [61](#)

## T

Text, [62](#)

## U

UserForm\_Activate, [62](#)

## V

Value, [61](#), [62](#)

## X

X, [61](#)

## Y

Y, [61](#)

```
1
2 Option Explicit
3
4
5 Private Sub CloseBtn_Click()
6     LoginForm.Hide
7     'ThisWorkbook.Close False
8 End Sub
9
10 Private Sub OkBtn_Click()
11     CheckUser
12 End Sub
13
14
15 Private Sub PasswordField_Exit(ByVal Cancel As MSForms.ReturnBoolean)
16     Sheet14.Range("B6" ).Value = PasswordField.Value
17 End Sub
18
19 Private Sub UserForm_Initialize()
20     HideBar Me 'Remove Border and Title Bar
21 End Sub
22
23
24
25 Private Sub UserNameField_Exit(ByVal Cancel As MSForms.ReturnBoolean)
26     Sheet14.Range("B5" ).Value = UserNameField.Value
27 End Sub
```

## C

Cancel, [64](#)  
CheckUser, [64](#)  
CloseBtn\_Click, [64](#)

## E

Explicit, [64](#)

## H

Hide, [64](#)  
HideBar, [64](#)

## L

LoginForm, [64](#)

## M

MSForms, [64](#)

## O

OkBtn\_Click, [64](#)

## P

PasswordField, [64](#)  
PasswordField\_Exit, [64](#)

## R

Range, [64](#)  
ReturnBoolean, [64](#)

## S

Sheet14, [64](#)

## U

UserForm\_Initialize, [64](#)  
UserNameField, [64](#)  
UserNameField\_Exit, [64](#)

## V

Value, [64](#)

```
1 Option Explicit
2
3 Sub Notif_AddEmailAttach()
4     Dim FileFldr As FileDialog
5     Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
6     With FileFldr
7         .Title = "Select file to attach"
8         .Filters.Add "All Files" , "*.*"
9         If .Show <> -1 Then GoTo NoSelection
10        Sheet4.Range("G508" ).Value = .SelectedItems(1)
11    End With
12    NoSelection:
13 End Sub
14 Sub Notif_AddNew()
15     With Sheet4
16         .Range("B506,B508" ).Value = True 'Set Notification Load and New Notification To True
17         .Range("B507,E507,E508,G507,G508,E509:G509,D511:G517,I508:J513" ).ClearContents
18         .Shapes("ExistNotifGrp" ).Visible = msoFalse
19         .Shapes("NewNotifGrp" ).Visible = msoCTrue
20         .Range("B506" ).Value = False 'Set Notification Load To False
21         .Range("E507" ).Select
22     End With
23 End Sub
24 Sub Notif_CancelNew()
25     If Sheet4.Range("D521" ).Value <> Empty Then Sheet4.Range("D521" ).Select
26 End Sub
27
28 Sub Notif_Delete()
29     Dim NotifRow As Long
30     Dim LastRow As Long
31     With Sheet4
32         If .Range("B507" ).Value = Empty Then Exit Sub 'Exit on empty row
33         If MsgBox("Are you sure you want to delete this notification?" , vbYesNo, "Delete
Notification" ) = vbNo Then Exit Sub
34         NotifRow = .Range("B507" ).Value 'Notification Row
35         .Range("D" & NotifRow & ":L" & NotifRow).ClearContents
36         LastRow = .Range("D541" ).End(xlUp).Row
37         If LastRow > NotifRow Then 'If The Cleared Row is NOT the last row
38             .Range("D" & NotifRow & ":L" & LastRow - 1).Value = .Range("D" & NotifRow + 1 &
":L" & LastRow).Value 'Move all table contents up 1 row
39             .Range("D" & LastRow & ":L" & LastRow).ClearContents 'Clear Last Row
40         End If
41     End With
42 End Sub
43 Sub Notif_Load()
44     With Sheet4
45         Dim NotifCol As Long
46         Dim NotifRow As Long
47         If .Range("B507" ).Value = Empty Then Exit Sub 'Exit on empty row
48         NotifRow = .Range("B507" ).Value 'Set Notification Row
49         For NotifCol = 4 To 11
50             If NotifCol <> 10 Then .Range(.Cells(542, NotifCol).Value).Value = .Cells(NotifRow,
NotifCol).Value 'Load Row Values using data mapping
51         Next NotifCol
52         .Range("B508" ).Value = False 'Set New Notification To False
53         .Shapes("ExistNotifGrp" ).Visible = msoCTrue
54         .Shapes("NewNotifGrp" ).Visible = msoFalse
55     End With
56 End Sub
57
58 Sub Notif_Save()
```

1



```
1
59 With Sheet4
60     Dim NotifRow As Long
61     Dim NotifCol As Long
62     'Check to Make sure an existing event type
63     If .Range("G507").Value = Empty Then
64         MsgBox "Please enter an Event Type to Save this Notification"
65         .Range("G507").Select
66         Exit Sub
67     End If
68     If .Range("B508").Value = True Then 'New Notification
69         If .Range("B509").Value = "Existing" Then 'Check For Existing
70             MsgBox "This Event type already has a notification. Please select that
71                 notification to edit it"
72             Exit Sub
73         End If
74         NotifRow = .Range("D541").End(xlUp).Row + 1 'First Avail. Row
75     Else: 'Existing
76         NotifRow = .Range("B507").Value 'Notification Row
77     End If
78     'Add Data to the table below
79     For NotifCol = 4 To 11
80         If NotifCol <> 10 Then .Cells(NotifRow, NotifCol).Value = .Range(.Cells(542,
81             NotifCol).Value).Value
82         Next NotifCol
83     .Shapes("ExistNotifGrp").Visible = msoCTrue
84     .Shapes("NewNotifGrp").Visible = msoFalse
85     .Range("B508").Value = False 'Set New Notification To False
End With
End Sub
```

## A

Add, [66](#)  
Application, [66](#)

## C

Cells, [66](#), [67](#)  
ClearContents, [66](#)

## E

Empty, [66](#), [67](#)  
Explicit, [66](#)

## F

FileDialog, [66](#)  
FileFldr, [66](#)  
Filters, [66](#)

## L

LastRow, [66](#)

## M

MsgBox, [66](#), [67](#)  
msoCTrue, [66](#), [67](#)  
msoFalse, [66](#), [67](#)  
msoFileDialogFilePicker, [66](#)

## N

NoSelection, [66](#)  
Notif\_AddEmailAttach, [66](#)  
Notif\_AddNew, [66](#)  
Notif\_CancelNew, [66](#)  
Notif\_Delete, [66](#)  
Notif\_Load, [66](#)  
Notif\_Save, [66](#)  
NotifCol, [66](#), [67](#)  
NotifRow, [66](#), [67](#)

## R

Range, [66](#), [67](#)  
Row, [66](#), [67](#)

## S

SelectedItems, [66](#)  
Shapes, [66](#), [67](#)  
Sheet4, [66](#), [67](#)  
Show, [66](#)

## T

Title, [66](#)

## V

Value, [66](#), [67](#)  
vbNo, [66](#)  
vbYesNo, [66](#)  
Visible, [66](#), [67](#)

## X

xUp, [66](#), [67](#)

```

1 Option Explicit
2 Dim TimeInteger As Long
3 Sub Admin_SetTotalFormat()
4     If Sheet4.Range("TimeFormat" ).Value = "Decimal (2.50)" Then
5         Sheet1.Range("G28:I44" ).NumberFormat = "0.00" 'Employee Manager
6         Sheet16.Range("AI14:AJ44" ).NumberFormat = "0.00" 'Payroll
7         Sheet15.Range("AB17,BF7:BH100" ).NumberFormat = "0.00" 'Reports
8     Else: 'Time Format
9         Sheet1.Range("G28:I44" ).NumberFormat = "h:mm;@" 'Employee Manager sheet
10        Sheet16.Range("AI14:AJ44" ).NumberFormat = "h:mm;@" 'Payroll Sheet
11        Sheet15.Range("AB17,BF7:BH100" ).NumberFormat = "h:mm;@" 'Reports
12    End If
13 End Sub
14
15 Sub BrowseMainApplicationFolder()
16     Dim EmpFldr As FileDialog
17     Dim strPath As String
18     Set EmpFldr = Application.FileDialog(msoFileDialogFolderPicker)
19     With EmpFldr
20         .Title = "Select a Main Application Folder"
21         If .Show <> -1 Then GoTo NoSelection
22         Sheet4.Range("J16" ).Value = .SelectedItems(1)
23     End With
24     NoSelection:
25 End Sub
26 Sub CheckMainAppFolder()
27     'Check for missing/ incorrect Application folder
28     On Error Resume Next
29     If Dir(Sheet4.Range("ApplicationFolder" ).Value, vbDirectory) = "" Then
30         Sheet4.Activate
31         Sheet4.Range("D4" ).Select 'Select First tab
32         Sheet4.Range("J16" ).Select
33         MsgBox "Plese set the correct Application Folder where the Attachments folder will be
34             stored"
35         BrowseMainApplicationFolder
36     End If
37 End Sub
38
39
40 Sub CreatePayPeriods()
41     Dim PayrollQty As Long
42     Dim StartRow As Long
43     Dim FormulaRow As Long
44     Dim StartDate As Date
45     Dim NextDate As Date
46     Dim Frequency As String
47     With Sheet4
48         If .Range("B4" ).Value = Empty Or .Range("B5" ).Value = Empty Or .Range("B6" ).Value =
49             Empty Then
50             MsgBox "Please make sure that all fields have been filled in before Creating Pay
51                 Periods"
52             Exit Sub
53         End If
54         CreatePayPeriod.Hide
55         Frequency = .Range("B4" ).Value 'Frequency
56         StartDate = .Range("B5" ).Value 'Start Date
57         PayrollQty = .Range("B6" ).Value 'Payroll Qty
58
59         Select Case Frequency
60             Case Is = "Weekly"

```

```
1 2 3
59     FormulaRow = 15
60     Case Is = "Bi-Weekly"
61         FormulaRow = 17
62     Case Is = "Semi-Monthly"
63         FormulaRow = 19
64     Case Is = "Monthly"
65         FormulaRow = 21
66     End Select
67
68     If .Range("B7").Value = True Then 'New List
69         .Range("D209:F304").ClearContents
70         StartRow = 209
71     Else:
72         StartRow = .Range("D304").End(xlUp).Row + 1 'First Available Row
73     End If
74     .Range("E" & StartRow).Value = StartDate
75
76     .Range("B" & FormulaRow).Copy
77     .Range("F" & StartRow & ":F" & StartRow + PayrollQty - 1).PasteSpecial
78     xlPasteFormulas
79     .Range("A" & FormulaRow + 1).Copy
80     .Range("E" & StartRow + 1 & ":E" & StartRow + PayrollQty - 1).PasteSpecial
81     xlPasteFormulas
82     .Range("B14").Copy
83     .Range("D" & StartRow & ":D" & StartRow + PayrollQty - 1).PasteSpecial
84     xlPasteFormulas 'Create Pay Period Names
85     End With
86     Application.CutCopyMode = False
87 End Sub
88
89 Sub ShowPayPeriodForm()
90     CreatePayPeriod.Show
91 End Sub
```

**A**

Activate, 69  
Admin\_SetTotalFormat, 69  
Application, 69, 70

**B**

BrowseMainApplicationFolder, 69

**C**

CheckMainAppFolder, 69  
ClearContents, 70  
Copy, 70  
CreatePayPeriod, 69, 70  
CreatePayPeriods, 69  
CutCopyMode, 70

**D**

Dir, 69

**E**

EmpFldr, 69  
Empty, 69  
Explicit, 69

**F**

FileDialog, 69  
FormulaRow, 69, 70  
Frequency, 69

**H**

Hide, 69

**M**

MsgBox, 69  
msoFileDialogFolderPicker, 69

**N**

NextDate, 69  
NoSelection, 69  
NumberFormat, 69

**P**

PasteSpecial, 70  
PayrollQty, 69, 70

**R**

Range, 69, 70  
Row, 70

**S**

SelectedItems, 69  
Sheet1, 69  
Sheet15, 69  
Sheet16, 69  
Sheet4, 69  
Show, 69, 70  
ShowPayPeriodForm, 70  
StartDate, 69, 70  
StartRow, 69, 70  
strPath, 69

**T**

TimeInteger, 69  
Title, 69

**V**

Value, 69, 70  
vbDirectory, 69

**X**

xlPasteFormulas, 70  
xlUp, 70

```
1 Option Explicit
2
3 Sub Admin_SwitchHorizontalTabs()
4     Dim SelCol As Long
5     Dim FirstRow As Long
6     StopCalc
7     With Sheet4
8         SelCol = .Range("B3").Value
9         If SelCol = 6 Then .Shapes("CreatePayPeriodBtn").Visible = msoCTrue Else: .Shapes(
10            "CreatePayPeriodBtn").Visible = msoFalse
11        If SelCol = 4 Then .Shapes("BrowseFolderBtn").Visible = msoCTrue Else: .Shapes(
12            "BrowseFolderBtn").Visible = msoFalse
13        If SelCol = 9 Then
14            .Shapes("NotifGrp").Visible = msoCTrue
15            .Shapes("ExistNotifGrp").Visible = msoCTrue
16        Else:
17            .Shapes("NotifGrp").Visible = msoFalse
18            .Shapes("ExistNotifGrp").Visible = msoFalse
19        End If
20        .Shapes("NewNotifGrp").Visible = msoFalse 'This Group is hidden regardless unless the
21        user clicks the New Notification button
22        .Range("5:604").EntireRow.Hidden = True
23        FirstRow = 5 + ((SelCol - 4) * 100)
24        .Range(FirstRow & ":" & FirstRow + 99).EntireRow.Hidden = False
25    End With
26    ResetCalc
27 End Sub
```

**A**

Admin\_SwitchHorizontalTabs, [72](#)

**E**

EntireRow, [72](#)

Explicit, [72](#)

**F**

FirstRow, [72](#)

**H**

Hidden, [72](#)

**M**

msoCTrue, [72](#)

msoFalse, [72](#)

**R**

Range, [72](#)

ResetCalc, [72](#)

**S**

SelCol, [72](#)

Shapes, [72](#)

Sheet4, [72](#)

StopCalc, [72](#)

**V**

Value, [72](#)

Visible, [72](#)

```

1  Option Explicit
2
3  Sub SetSyncFrequency()
4      Dim MinNum
5      If Sheet4.Range("SharingMode" ).Value = "On" Then
6          MinNum = Sheet4.Range("B32" ).Value '# of Minutes to sync
7          Application.OnTime Now + TimeValue("00:" & MinNum & ":00" ), "UpdateLocal"
8      End If
9  End Sub
10
11 Sub UpdateLocal()
12     Sheet4.Range("B1" ).Value = True
13     Dim FileName As String
14     Dim FilePath As String
15     Dim LongFileName As String
16     Dim fso As Object
17     Dim oFile As Object
18     Dim TCFile As String
19     Dim SyncText As String
20     Dim SyncTextLine As String
21     Dim UserName As String
22     Dim SheetName As String
23     Dim CellAddress As String
24     Dim CellValue As String
25
26     If Sheet4.Range("SharingMode" ).Value <> "On" Then Exit Sub 'Exit when sharing mode is
off
27     Set fso = CreateObject("Scripting.FileSystemObject" ) 'Set File System Object
28     UserName = Sheet14.Range("AppUserName" ).Value 'Username
29     CheckMainAppFolder 'Run Macro to Check For Correct Main Application Folder
30     On Error GoTo NoFolder
31     If Dir(Sheet4.Range("ApplicationFolder" ).Value, vbDirectory) = "" Then Exit Sub
32     FilePath = Sheet4.Range("ApplicationFolder" ).Value & "\" & UserName & "\"
33     If Dir(FilePath, vbDirectory) = "" Then 'If folder does not exist, Create Folder & Exit
Sub
34         fso.createFolder (FilePath) 'Add Folder if needed
35         Exit Sub 'No Files in New Folder so we can exit
36     End If
37
38     Sheet4.Range("B1" ).Value = True 'Set Sync as True
39     FileName = Dir(FilePath & "*.txt" )
40     Do While Len(FileName) > 0 'Start of Loop
41         LongFileName = FilePath & FileName
42         Open LongFileName For Input As #1
43         Line Input #1, SyncText
44         Close #1
45         SheetName = Left(SyncText, InStr(SyncText, "," ) - 1)
46         CellAddress = Mid(SyncText, InStr(SyncText, "," ) + 1, InStr(SyncText, ":" ) - InStr(
SyncText, "," ) - 1)
47         CellValue = Right(SyncText, Len(SyncText) - InStr(SyncText, ":" ))
48         ThisWorkbook.Sheets(SheetName).Range(CellAddress).Value = CellValue
49         'Delete text file
50         Kill (LongFileName)
51         FileName = Dir() 'Clear Curent File Name
52     Loop 'End Of Loop
53
54     Set fso = Nothing
55     Set oFile = Nothing
56     Empl_SortByName 'Update & Sort Employee List
57     NoFolder:
58     Sheet4.Range("B1" ).Value = False

```

1



```
59 1 SetSyncFrequency 'Rerun Sync at specified interval
60 End Sub
61
62
63
```

**A**

Application, [74](#)

**C**

CellAddress, [74](#)  
CellValue, [74](#)  
CheckMainAppFolder, [74](#)  
Close, [74](#)  
createFolder, [74](#)  
CreateObject, [74](#)

**D**

Dir, [74](#)

**E**

Empl\_SortByName, [74](#)  
Explicit, [74](#)

**F**

FileName, [74](#)  
FilePath, [74](#)  
fso, [74](#)

**I**

Input, [74](#)  
InStr, [74](#)

**K**

Kill, [74](#)

**L**

Left, [74](#)  
Len, [74](#)  
Line, [74](#)  
LongFileName, [74](#)

**M**

Mid, [74](#)  
MinNum, [74](#)

**N**

NoFolder, [74](#)  
Now, [74](#)

**O**

oFile, [74](#)  
OnTime, [74](#)  
Open, [74](#)

**R**

Range, [74](#)  
Right, [74](#)

**S**

SetSyncFrequency, [74](#), [75](#)  
Sheet14, [74](#)  
Sheet4, [74](#)  
SheetName, [74](#)  
Sheets, [74](#)  
SyncText, [74](#)  
SyncTextLine, [74](#)

**T**

TCFile, [74](#)  
ThisWorkbook, [74](#)  
TimeValue, [74](#)

**U**

UpdateLocal, [74](#)

UserName, [74](#)

**V**

Value, [74](#)  
vbDirectory, [74](#)

```

1
2 Sub do_nothing()
3   'nothing to do
4 End Sub
5
6
7 Private Function GetColltr(sAddr As String) As String
8   Dim iPos As Long, sTest As String
9   Do While iPos < 3
10    iPos = iPos + 1
11    If IsNumeric(Mid(sAddr, iPos, 1)) Then
12      Exit Do
13    Else
14      sTest = sTest & Mid(sAddr, iPos, 1)
15    End If
16  Loop
17  GetColltr = sTest
18 End Function
19
20
21
22 Function GetTabOrder() As Variant
23   '--set the tab order of input cells - change ranges as required
24   ' don't include "$" in these cell references
25   If ActiveSheet.CodeName = "Sheet1" Then 'For Employee Mgr Sheet
26     If ActiveCell.Row < 19 Then GetTabOrder = Array("F6" , "I6" , "F8" , "F10" , "I10" ,
27       "F12" , "H12" , "F14" , "H14" , "K14" , "F16" , "H16" , "K16" )
28     If ActiveCell.Row = 26 Then GetTabOrder = Array("F26" , "I26" , "K26" )
29     If ActiveCell.Row = 46 Then GetTabOrder = Array("F46" , "I46" , "K46" )
30     If ActiveCell.Row > 65 And ActiveCell.Row < 76 Then GetTabOrder = Array("G66" , "J66" ,
31       "G68" , "J68" , "G70" , "J70" , "G73" , "J73" , "G75" , "J75" )
32     If ActiveCell.Row > 85 And ActiveCell.Row < 89 Then GetTabOrder = Array("G86" , "J86" ,
33       "G88" , "J88" )
34     If ActiveCell.Row > 164 And ActiveCell.Row < 184 Then GetTabOrder = Array("G182" ,
35       "I182" , "K182" )
36   End If
37   If ActiveSheet.CodeName = "Sheet5" Then 'Events Sheet
38     If ActiveCell.Row > 3 And ActiveCell.Row < 10 Then GetTabOrder = Array("E4" , "H4" ,
39       "K4" , "E6" )
40     If ActiveCell.Row = 10 Or ActiveCell.Row = 11 Then GetTabOrder = Array("F10" , "G10" ,
41       "J10" , "L10" , "N10" , "J11" , "L11" )
42     If ActiveCell.Row = 13 Then GetTabOrder = Array("F13" , "G13" , "I13" , "L13" )
43     If ActiveCell.Row > 14 And ActiveCell.Row < 22 Then GetTabOrder = Array("F15" , "K15" ,
44       "F16" , "F17" , "F18" )
45   End If
46   If ActiveSheet.CodeName = "Sheet4" Then 'Admin Sheet
47     If ActiveCell.Row > 506 And ActiveCell.Row < 518 Then GetTabOrder = Array("E507" ,
48       "G507" , "E508" , "G508" , "E509" , "D511" , "I508" )
49     If ActiveCell.Row > 407 And ActiveCell.Row < 428 Then GetTabOrder = Array("E408" ,
50       "F408" , "F409" , "E410" , "F410" , "E413" , "F413" , "F414" , "E415" , "F415" ,
51       "F418" , "F419" , "F422" , "F423" , "F424" , "F425" , "F426" , "F427" )
52   End If
53 End Function
54
55 Sub SetOnkey(ByVal state As Boolean)
56   'Compiled By Randy Austin
57   'Workbook Provided By www.ExcelForFreelancers.com
58   If state Then
59     With Application
60       .OnKey "{TAB}" , "'TabRange xlNext'"
61       .OnKey "~" , "'TabRange xlNext'"
62       .OnKey "{ENTER}" , "'TabRange xlNext'"
63     End With
64   End If
65 End Sub

```

1 2 3

```

52     1 2 3     .OnKey "{RIGHT}" , "'TabRange xlNext'"
53     .OnKey "{LEFT}" , "'TabRange xlPrevious'"
54     .OnKey "+{TAB}" , "'TabRange xlPrevious'"
55     .OnKey "{DOWN}" , "'UpOrDownArrow xlDown'"
56     .OnKey "{UP}" , "'UpOrDownArrow xlUp'"
57     End With
58     Else
59     'reset keys
60     With Application
61     .OnKey "{ENTER}"
62     .OnKey "{TAB}"
63     .OnKey "~"
64     .OnKey "+{TAB}"
65     .OnKey "{RIGHT}"
66     .OnKey "{LEFT}"
67     .OnKey "{DOWN}"
68     .OnKey "{UP}"
69     End With
70     End If
71 End Sub

72
73 Sub TabRange(Optional iDirection As Integer = xlNext)
74
75     Dim vTabOrder As Variant, m As Variant
76     Dim lItems As Long, iAdjust As Long
77     On Error GoTo ExitSub
78     '--get the tab order from shared function
79     vTabOrder = GetTabOrder
80     lItems = UBound(vTabOrder) - LBound(vTabOrder) + 1
81
82     On Error Resume Next
83     m = Application.Match(ActiveCell.Address(0, 0), vTabOrder, False)
84
85
86     '--if activecell is not in Tab Order return to the first cell
87     If IsError(m) Then
88         m = 1
89     Else
90         '--get adjustment to index
91         iAdjust = IIf(iDirection = xlPrevious, -1, 1)
92
93         '--calculate new index wrapping around list
94         m = (m + lItems + iAdjust - 1) Mod lItems + 1
95     End If
96
97     '--select cell adjusting for Option Base 0 or 1
98     Application.EnableEvents = False
99     Range(vTabOrder(m + (LBound(vTabOrder) = 0))).Select
100
101     ExitSub:
102     Application.EnableEvents = True
103
104
105 End Sub

106
107 Sub UpOrDownArrow(Optional iDirection As Integer = xlUp)
108
109     Dim vTabOrder As Variant
110     Dim lRowClosest As Long, lRowTest As Long
111     Dim i As Long, iSign As Integer
112

```

1

```
1
113 Dim sActiveCol As String
114 Dim bFound As Boolean
115
116 '--get the tab order from shared function
117 vTabOrder = GetTabOrder
118 On Error Resume Next
119 If vTabOrder = Empty Then Exit Sub
120 '--find TabCells in same column as ActiveCell in iDirection
121 '-- rTest will include ActiveCell
122
123 sActiveCol = GetColLtr(ActiveCell.Address(0, 0))
124
125 iSign = IIf(iDirection = xlDown, -1, 1)
126 lRowClosest = IIf(iDirection = xlDown, Rows.Count + 1, 0)
127
128 For i = LBound(vTabOrder) To UBound(vTabOrder)
129     If GetColLtr(CStr(vTabOrder(i))) = sActiveCol Then
130         lRowTest = Range(CStr(vTabOrder(i))).Row
131
132         '--find closest cell to ActiveCell in rTest
133         If iSign * lRowTest > iSign * lRowClosest And _
134             iSign * lRowTest < iSign * ActiveCell.Row Then
135             '--at least one cell in iDirection of same column
136             bFound = True
137             lRowClosest = lRowTest
138         End If
139     End If
140 Next i
141
142 If bFound Then
143     Application.EnableEvents = False
144     Cells(lRowClosest, ActiveCell.Column).Select
145     Application.EnableEvents = True
146 End If
147 End Sub
```

→, 79

## A

ActiveCell, 77-79  
ActiveSheet, 77  
Address, 78, 79  
Application, 77-79  
Array, 77

## B

bFound, 79

## C

Cells, 79  
CodeName, 77  
Column, 79  
Count, 79

## D

do\_nothing, 77

## E

Empty, 79  
EnableEvents, 78, 79  
ExitSub, 78

## G

GetColLtr, 77, 79  
GetTabOrder, 77-79

## I

i, 78, 79  
iAdjust, 78  
iDirection, 78, 79  
If, 78, 79  
iPos, 77  
IsError, 78  
iSign, 78, 79  
IsNumeric, 77

## L

LBound, 78, 79  
lItems, 78  
lRowClosest, 78, 79  
lRowTest, 78, 79

## M

m, 78  
Match, 78  
Mid, 77

## O

OnKey, 77, 78

## R

Range, 78, 79  
Row, 77, 79  
Rows, 79

## S

sActiveCol, 79  
sAddr, 77  
SetOnkey, 77  
state, 77  
sTest, 77

## T

TabRange, 78

## U

UBound, 78, 79  
UpOrDownArrow, 78

## V

vTabOrder, 78, 79

## X

xlDown, 79  
xlNext, 78  
xlPrevious, 78  
xlUp, 78

```
1 Option Explicit
2 Sub ResetCalc()
3     With Application
4         .EnableEvents = True
5         .Calculation = xlCalculationAutomatic
6         .ScreenUpdating = True
7     End With
8 End Sub
9
10
11 Sub StopCalc()
12     With Application
13         .EnableEvents = False
14         .Calculation = xlCalculationManual
15         .ScreenUpdating = False
16     End With
17 End Sub
```

**A**

Application, [81](#)

**C**

Calculation, [81](#)

**E**

Explicit, [81](#)

**R**

ResetCalc, [81](#)

**S**

ScreenUpdating, [81](#)

StopCalc, [81](#)

**X**

xlCalculationAutomatic, [81](#)

xlCalculationManual, [81](#)



```
1 Option Explicit
2 Dim StartTime As Double
3 Dim SecondsElapsed As Double
4
5 Sub CalculateRunTime_Minutes()
6 'PURPOSE: Determine how many minutes it took for code to completely run
7 'SOURCE: www.TheSpreadsheetGuru.com/the-code-vault
8
9 Dim StartTime As Double
10 Dim MinutesElapsed As String
11
12 'Remember time when macro starts
13 StartTime = Timer
14
15 '*****
16 'Insert Your Code Here...
17 '*****
18
19 'Determine how many seconds code took to run
20 MinutesElapsed = Format((Timer - StartTime) / 86400, "hh:mm:ss" )
21
22 'Notify user in seconds
23 MsgBox "This code ran successfully in " & MinutesElapsed & " minutes" , vbInformation
24
25 End Sub
26 Sub StartClock()
27
28 'Remember time when macro starts
29 StartTime = Timer
30 End Sub
31
32 Sub StopClock()
33
34 'Determine how many seconds code took to run
35 SecondsElapsed = Round(Timer - StartTime, 2)
36
37 'Notify user in seconds
38 MsgBox "This code ran successfully in " & SecondsElapsed & " seconds" , vbInformation
39 End Sub
```

**C**

CalculateRunTime\_Minutes, [83](#)

**E**

Explicit, [83](#)

**F**

Format, [83](#)

**M**

MinutesElapsed, [83](#)

MsgBox, [83](#)

**R**

Round, [83](#)

**S**

SecondsElapsed, [83](#)

StartClock, [83](#)

StartTime, [83](#)

StopClock, [83](#)

**T**

Timer, [83](#)

**V**

vbInformation, [83](#)

```

1  Option Explicit
2
3  Sub Attach_Add()
4      Dim FileFldr As FileDialog
5      Dim FileName As String
6      Dim OrigFilePath As String
7      Dim FileType As String
8      Dim MainAppFolder As String
9      Dim DestFileFolder As String
10     Dim DestFilePath As String
11     Dim EmpID As String
12     Dim UserName As String
13     Dim LastAttRow As Long
14
15     CheckMainAppFolder 'Run Macro to Check For Correct Main Application Folder
16
17     If Dir(Sheet4.Range("ApplicationFolder" ).Value, vbDirectory) = "" Then Exit Sub 'Exit
on still no correct folder
18     MainAppFolder = Sheet4.Range("ApplicationFolder" ).Value 'Set Main Application Folder
19     If Sheet1.Range("B6" ).Value = True Then Empl_SaveNew 'Save Employee to Lock In Employee
ID
20
21     EmpID = Sheet1.Range("J2" ).Value 'Employee ID
22
23     Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
24     With FileFldr
25         .Title = "Select file to attach"
26         .Filters.Add "All Files" , "*.*"
27         If .Show <> -1 Then GoTo NoSelection
28         OrigFilePath = .SelectedItems(1)
29         FileName = Dir(OrigFilePath)
30
31         FileType = Right(FileName, 5 - InStr(1, Right(FileName, 5), "." ))
32     End With
33
34
35
36     'Check For Attachments Folder and add if needed
37     If Dir(MainAppFolder & "/Attachments" , vbDirectory) = "" Then 'If attach folder does
not exist, Create Folder & Exit Sub
38     MkDir MainAppFolder & "/Attachments" 'Add Folder if needed
39     End If
40
41     'Check For Employee Folder and Add if needed
42     DestFileFolder = MainAppFolder & "\Attachments\" & EmpID & "_Emp_ID" 'Set Employee
Attachment Folder
43     If Dir(DestFileFolder, vbDirectory) = "" Then 'If Employee Folder does not Exist, then
create it
44     MkDir DestFileFolder 'Add Folder if needed
45     End If
46
47
48     DestFilePath = DestFileFolder & "\" & FileName
49     FileCopy OrigFilePath, DestFilePath 'Copy user selected file into appropriate Employee
folder
50     'Kill (OrigFilePath) 'If you want to delete the original file in the original location
51
52     If Sheet4.Range("J17" ).Value = "On" Then UserName = Sheet14.Range("AppUserName" ).Value
Else: UserName = Application.UserName
53     With Sheet3
54         LastAttRow = .Range("D9999" ).End(xlUp).Row + 1 'First Avail. Attachment Row

```

```

1 2
55 .Range("D" & LastAttRow).Value = EmpID 'EmployeeID
56 .Range("E" & LastAttRow).Value = Sheet1.Range("F2" ).Value 'Employee Name
57 .Range("F" & LastAttRow).Value = FileName
58 .Range("G" & LastAttRow).Value = FileType
59 .Range("H" & LastAttRow).Value = FileName
60 .Range("I" & LastAttRow).Value = UserName
61 .Range("J" & LastAttRow).Value = Now
62 .Range("K" & LastAttRow).Value = "=Row()"
63 End With
64
65 Attach_Refresh 'Refresh List To Display New Attachment
66 Sheet1.Range("E19" ).Select 'Select First Item to Display Thumbnail (if any)
67 Attach_DisplayThumbnail
68 NoSelection:
69 End Sub
70 Sub Attach_Delete()
71 Dim EmpID, AttRow As Long
72 If Sheet1.Range("J" & ActiveCell.Row).Value = Empty Then
73 MsgBox "Please select an Attachment to delete first"
74 Exit Sub
75 End If
76 On Error Resume Next
77 Sheet1.Shapes("AttachThumb" ).Delete
78 On Error GoTo 0
79
80 AttRow = Sheet1.Range("J" & ActiveCell.Row).Value 'Attachement Row
81 Sheet3.Range("D" & AttRow).ClearContents 'clear just the Employee ID so it will not
appear in Employe section
82 Sheet3.Range("H" & AttRow).Value = "Removed By: " & Sheet14.Range("AppUserName" ).Value
83 Attach_Refresh
84 Sheet1.Range("E19" ).Select
85 Attach_DisplayThumbnail
86 End Sub
87
88 Sub Attach_DisplayThumbnail()
89 Dim AttRow As Long
90 Dim AttTableRow As Long
91 Dim FileType As String
92 Dim FilePath As String
93 Dim ThumbType As String
94 Dim EmpID As String
95 Dim MainAppFolder As String
96 With Sheet1
97 On Error Resume Next
98 .Shapes("AttachThumb" ).Delete
99 On Error GoTo 0
100 If .Range("B7" ).Value = Empty Then Exit Sub 'Exit on No AttachmentRow
101 CheckMainAppFolder 'Run Macro to Check For Correct Main Application Folder
102 If Dir(Sheet4.Range("ApplicationFolder" ).Value, vbDirectory) = "" Then Exit Sub
'Exit on still no correct folder
103 MainAppFolder = Sheet4.Range("ApplicationFolder" ).Value 'Set Main Application Folder
104 If Sheet1.Range("B6" ).Value = True Then Empl_SaveNew 'Save Employee to Lock In
Employee ID
105
106 AttRow = .Range("B7" ).Value 'Set Attachment Row
107 EmpID = .Range("J2" ).Value 'Employee ID
108 FileType = LCase(.Range("F" & AttRow).Value)
109 AttTableRow = .Range("J" & AttRow).Value 'Attachement Table Row
110 FilePath = MainAppFolder & "\Attachments\" & EmpID & "_Emp_ID\" & .Range("G" &
AttRow).Value 'Full Attachment File Path
111 If FileType = "jpg" Or FileType = "png" Or FileType = "gif" Or FileType = "jpeg" Or

```

1 2 3

```

1 2 3
File Type = "bmp" Then 'Picture Types
112 On Error GoTo NoAttach
113 .Pictures.Insert(FilePath).Name = "AttachThumb"
114 With .Shapes("AttachThumb" )
115 .Visible = msoCTrue
116 .LockAspectRatio = True
117 .Left = Sheet1.Range("J19" ).Left
118 .Top = Sheet1.Range("J19" ).Top
119 If .Width > .Height Then .Width = 90 Else: .Height = 80
120 .IncrementLeft 30
121 .IncrementTop 2
122 End With
123 'On Word, Excel or PDF Section
124 ElseIf File Type = "docx" Or File Type = "pdf" Or File Type = "doc" Then
125 On Error Resume Next
126 .OLEObjects.Add(FileName:=FilePath, Link:=False, DisplayAsIcon:=False).Name =
"AttachThumb"
127 On Error GoTo NoAttach
128 With .Shapes("AttachThumb" )
129 .LockAspectRatio = True
130 .Left = Sheet1.Range("L4" ).Left
131 .Top = Sheet1.Range("L4" ).Top
132 If .Width > .Height Then .Width = 300 Else: .Height = 500
133 .IncrementLeft 35
134 .IncrementTop 2
135 .Visible = msoCTrue
136 End With
137
138 End If
139
140 End With
141 NoAttach:
142 End Sub
143 Sub Attach_Filter()
144 Dim LastAttRow As Long
145 With Sheet3
146 .Activate
147 .Range("AA2:AH2" ).ClearContents 'Clear Old Criteria
148 If .Range("D3" ).Value <> "Enter ID:" And .Range("D3" ).Value <> Empty Then .Range(
"AA2" ).Value = .Range("D3" ).Value Else: .Range("AA2" ).ClearContents 'Employee ID
149 If .Range("E3" ).Value <> "Enter Name:" And .Range("E3" ).Value <> Empty Then .Range(
"AB2" ).Value = "*" & .Range("E3" ).Value & "*" Else: .Range("AB2" ).ClearContents
'Employee Name
150 LastAttRow = .Range("D9999" ).End(xlUp).Row
151 If LastAttRow < 5 Then GoTo NoData
152 .Range("D4:K" & LastAttRow).AdvancedFilter xlFilterInPlace, CriteriaRange:=.Range(
"AA1:AH2" ), Unique:=True
153 NoData:
154 End With
155 End Sub
156
157 Sub Attach_FilterClear()
158 With Sheet3
159 On Error Resume Next
160 .ShowAllData
161 On Error GoTo 0
162 .Range("B3" ).Value = True 'Set Filter to True
163 .Range("D3" ).Value = "Enter ID:"
164 .Range("E3" ).Value = "Enter Name:"
165 .Range("B3" ).Value = False 'Set Filter to False
166 .Range("D5:D9999" ).EntireRow.Hidden = False

```

1 2

```

1 2
167 End With
168 End Sub
169 Sub Attach_Open()
170 With Sheet1
171 Dim AttRow As Long
172 Dim EmpID As String
173 Dim MainAppFolder As String
174 Dim FilePath As String
175
176 If .Range("B7").Value = Empty Then
177     MsgBox "Please select an attachment to open"
178     Exit Sub
179 End If
180 CheckMainAppFolder 'Run Macro to Check For Correct Main Application Folder
181 If Dir(Sheet4.Range("ApplicationFolder").Value, vbDirectory) = "" Then Exit Sub
182 MainAppFolder = Sheet4.Range("ApplicationFolder").Value 'Set Main Application Folder
183 EmpID = Sheet1.Range("J2").Value 'Employee ID
184
185 AttRow = .Range("B7").Value 'Attachment Row
186 FilePath = MainAppFolder & "\Attachments\" & EmpID & "_Emp_ID\" & .Range("G" &
187 AttRow).Value 'Full Attachment File Path
188 If Dir(FilePath) = "" Then Exit Sub
189 ThisWorkbook.FollowHyperlink (FilePath)
190 End With
191 End Sub
192 Sub Attach_Refresh()
193 Dim EmpID, AttRow, LastAttRow, LastFiltRow, FiltRow As Long
194
195 EmpID = Sheet1.Range("J2").Value 'Employee ID
196 Sheet1.Range("E19:J24").ClearContents 'Clear Old Attachemets
197 AttRow = 19 'Set Starting Attachments Row
198 With Sheet3
199     .Range("AA2:AH2,AA4:AH9999").ClearContents 'Clear Prev Adv. Filter
200     .Range("AA2").Value = EmpID 'Set Empl. ID as Filter
201     LastAttRow = .Range("D9999").End(xlUp).Row
202     If LastAttRow < 5 Then GoTo NoData
203     .Range("D4:K" & LastAttRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
204 "AA1:AH2"), CopyToRange:=.Range("AA3:AG3"), Unique:=True
205 LastFiltRow = .Range("AA9999").End(xlUp).Row 'Last Filtered Row
206 If LastFiltRow < 4 Then Exit Sub 'If no Attach. found then Exit
207 For FiltRow = LastFiltRow To 4 Step -1 'Use Reverse For next to put most recent
208     Attach. First
209     Sheet1.Range("E" & AttRow & ":J" & AttRow).Value = .Range("AB" & FiltRow & ":AG"
210 & FiltRow).Value 'Copy Over Attachment Info
211     AttRow = AttRow + 1
212     If AttRow = 25 Then Exit Sub 'Exit If Table is Full
213 Next FiltRow
214 End With
215 NoData:
216 End Sub
217 Sub Attach_ShowAll()
218 Sheet3.Range("D3").Value = Sheet1.Range("J2").Value 'Employee ID
219 End Sub

```

**A**

Activate, 87  
ActiveCell, 86  
Add, 85, 87  
AdvancedFilter, 87, 88  
Application, 85  
Attach\_Add, 85  
Attach\_Delete, 86  
Attach\_DisplayThumbnail, 86  
Attach\_Filter, 87  
Attach\_FilterClear, 87  
Attach\_Open, 88  
Attach\_Refresh, 86, 88  
Attach\_ShowAll, 88  
AttRow, 86, 88  
AttTableRow, 86

**C**

CheckMainAppFolder, 85, 86, 88  
ClearContents, 86-88  
CopytoRange, 88  
CriteriaRange, 87, 88

**D**

Delete, 86  
DestFileFolder, 85  
DestFilePath, 85  
Dir, 85, 86, 88  
DisplayAsIcon, 87

**E**

EmpID, 85, 86, 88  
Empl\_SaveNew, 85, 86  
Empty, 86-88  
EntireRow, 87  
Explicit, 85

**F**

FileCopy, 85  
FileDialog, 85  
FileFldr, 85  
FileName, 85-87  
FilePath, 86-88  
FileType, 85-87  
Filters, 85  
FiltRow, 88  
FollowHyperlink, 88

**H**

Height, 87  
Hidden, 87

**I**

IncrementLeft, 87  
IncrementTop, 87  
Insert, 87  
InStr, 85

**L**

LastAttRow, 85-88  
LastFiltRow, 88  
LCase, 86  
Left, 87  
Link, 87  
LockAspectRatio, 87

**M**

MainAppFolder, 85, 86, 88  
MkDir, 85

MsgBox, 86, 88  
msoCTrue, 87  
msoFileDialogFilePicker, 85

**N**

Name, 87  
NoAttach, 87  
NoData, 87, 88  
NoSelection, 85, 86  
Now, 86

**O**

OLEObjects, 87  
OrigFilePath, 85

**P**

Pictures, 87

**R**

Range, 85-88  
Right, 85  
Row, 85-88

**S**

SelectedItems, 85  
Shapes, 86, 87  
Sheet1, 85-88  
Sheet14, 85, 86  
Sheet3, 85-88  
Sheet4, 85, 86, 88  
Show, 85  
ShowAllData, 87

**T**

ThisWorkbook, 88  
ThumbType, 86  
Title, 85  
Top, 87

**U**

Unique, 87, 88  
UserName, 85, 86

**V**

Value, 85-88  
vbDirectory, 85, 86, 88  
Visible, 87

**W**

Width, 87

**X**

xlFilterCopy, 88  
xlFilterInPlace, 87  
xlUp, 85, 87, 88

```

1 Option Explicit
2 Dim ActCol As Long
3 Dim ActRow As Long
4 Dim TextBoxName As String
5 Sub EmpIDCard_CreateTextBox()
6     With Sheet1
7         .Shapes.AddTextbox(msoTextOrientationHorizontal, 2850, 162, 93, 20).Select
8         With Selection
9             If ActCol = 57 Or ActCol = 58 Then .Formula = "=" & Sheet1.Cells(ActRow, 55).
                Address Else: .Text = Sheet1.Range("BB" & ActRow).Value
10            .Name = "Box " & Sheet1.Cells(6, ActCol).Value & Sheet1.Range("BA" & ActRow).Value
11            TextBoxName = "Box " & Sheet1.Cells(6, ActCol).Value & Sheet1.Range("BA" & ActRow).
                Value
12        End With
13        With .Shapes(TextBoxName)
14            .Fill.Visible = msoFalse
15            .Line.Visible = msoFalse
16            .TextFrame2.TextRange.ParagraphFormat.Alignment = msoAlignCenter
17        End With
18    End With
19 End Sub
20 Sub EmpIDCard_Customize()
21     With Sheet1
22         .Shapes("CustomIDBtn" ).Visible = msoFalse
23         .Shapes("DoneCustomIDBtn" ).Visible = msoCTrue
24         .Range("BA:BF" ).EntireColumn.Hidden = False
25     End With
26 End Sub
27
28 Sub EmpIDCard_FieldChange()
29     With Sheet1
30         Dim TextBox As Shape
31         ActCol = ActiveCell.Column
32         ActRow = ActiveCell.Row
33         'On Selection of a Visible Field
34         If ActRow = 10 And ActCol > 56 Then 'Picture File so Do Nothing
35         Else: 'Non Picture File
36             TextBoxName = "Box " & .Cells(6, ActCol).Value & .Range("BA" & ActRow).Value
37             On Error Resume Next
38             Set TextBox = .Shapes(TextBoxName)
39             If TextBox Is Nothing Then EmpIDCard_CreateTextBox
40         End If
41         If ActiveCell.Value = "ü" Then
42             If ActRow = 10 And ActCol > 56 Then EmpIDCard_HidePic Else: .Shapes(TextBoxName).
                Visible = msoFalse 'If Picture, show picture, otherwise show field/label
43             ActiveCell.Value = " " 'Add Space so cell to the left does not run into this cell
44         Else: 'On Selection of a Hidden Field
45             If ActRow = 10 And ActCol > 56 Then EmpIDCard_ShowPic Else: .Shapes(TextBoxName).
                Visible = msoCTrue
46             ActiveCell.Value = "ü"
47         End If
48         .Range("BH4" ).Select
49     End With
50 End Sub
51 Sub EmpIDCard_Hide()
52     ActiveWindow.ScrollColumn = 3
53 End Sub
54
55 Sub EmpIDCard_HideCustom()
56     With Sheet1

```



```

1 2
57     .Shapes("CustomIDBtn" ).Visible = msoCTrue
58     .Shapes("DoneCustomIDBtn" ).Visible = msoFalse
59     .Range("BA:BF" ).EntireColumn.Hidden = True
60     End With
61 End Sub
62
63 Sub EmpIDCard_HidePic()
64     On Error Resume Next
65     Sheet1.Shapes("EmpIDPic" & Sheet1.Cells(6, ActCol).Value).Delete
66     On Error GoTo 0
67 End Sub
68 Sub EmpIDCard_Print()
69     Sheet1.PageSetup.PrintArea = "$BH$5:$BH$22"
70     ActiveSheet.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False
71 End Sub
72
73 Sub EmpIDCard_Show()
74     Sheet1.Range("E4" ).Select
75     'If Employee Picture has been displayed then run macro to update picture
76     If Sheet1.Range("BE10" ).Value = "ü" Then
77         ActCol = 57 'Set the Active Column to 57 to Display the front Picture
78         EmpIDCard_ShowPic
79     End If
80     If Sheet1.Range("BF10" ).Value = "ü" Then
81         ActCol = 58 'Set the Active Column to 57 to Display the front Picture
82         EmpIDCard_ShowPic
83     End If
84     ActiveWindow.ScrollColumn = 52
85 End Sub
86 Sub EmpIDCard_ShowPic()
87     With Sheet1
88         ' If .Range("BC10").Value = 0 Then GoTo NoPic
89         Dim PicPath As String
90         On Error Resume Next
91         .Shapes("EmpIDPic" & .Cells(6, ActCol).Value).Delete 'Delete Picture if it exists
92         On Error GoTo 0
93         PicPath = .Range("BC10" ).Value 'Path of the picture
94         If PicPath = "0" Then Exit Sub
95         On Error GoTo NoPic
96         With .Pictures.Insert(PicPath)
97             With .ShapeRange
98                 .LockAspectRatio = msoTrue
99                 .Height = 65
100                .AutoShapeType = msoShapeRoundedRectangle
101                .Name = "EmpIDPic" & Sheet1.Cells(6, ActCol).Value 'Add "Front" or "Back" to
                Picture Name
102            End With 'Shape Range
103        End With 'Pictures
104
105        With .Shapes("EmpIDPic" & .Cells(6, ActCol).Value) 'Add "Front" or "Back" to Picture
            Name
106            .Left = Sheet1.Range("BH6" ).Left
107            If ActCol = 57 Then .Top = Sheet1.Range("BH6" ).Top Else: .Top = Sheet1.Range(
                "BH16" ).Top
108            .IncrementLeft 180
109            .IncrementTop 15
110        End With
111        Exit Sub
112    NoPic:
113    End With 'End With Sheet 1
114 End Sub

```

**A**

ActCol, [90](#), [91](#)  
ActiveCell, [90](#)  
ActiveSheet, [91](#)  
ActiveWindow, [90](#), [91](#)  
ActRow, [90](#)  
Address, [90](#)  
AddTextbox, [90](#)  
Alignment, [90](#)  
AutoShapeType, [91](#)

**C**

Cells, [90](#), [91](#)  
Collate, [91](#)  
Column, [90](#)  
Copies, [91](#)

**D**

Delete, [91](#)

**E**

EmpIDCard\_CreateTextBox, [90](#)  
EmpIDCard\_Customize, [90](#)  
EmpIDCard\_FieldChange, [90](#)  
EmpIDCard\_Hide, [90](#)  
EmpIDCard\_HideCustom, [90](#)  
EmpIDCard\_HidePic, [90](#), [91](#)  
EmpIDCard\_Print, [91](#)  
EmpIDCard\_Show, [91](#)  
EmpIDCard\_ShowPic, [90](#), [91](#)  
EntireColumn, [90](#), [91](#)  
Explicit, [90](#)

**F**

Fill, [90](#)  
Formula, [90](#)

**H**

Height, [91](#)  
Hidden, [90](#), [91](#)

**I**

IgnorePrintAreas, [91](#)  
IncrementLeft, [91](#)  
IncrementTop, [91](#)  
Insert, [91](#)

**L**

Left, [91](#)  
Line, [90](#)  
LockAspectRatio, [91](#)

**M**

msoAlignCenter, [90](#)  
msoCTrue, [90](#), [91](#)  
msoFalse, [90](#), [91](#)  
msoShapeRoundedRectangle, [91](#)  
msoTextOrientationHorizontal, [90](#)  
msoTrue, [91](#)

**N**

Name, [90](#), [91](#)  
NoPic, [91](#)

**P**

PageSetup, [91](#)  
ParagraphFormat, [90](#)  
PicPath, [91](#)  
Pictures, [91](#)

PrintArea, [91](#)

PrintOut, [91](#)

**R**

Range, [90](#), [91](#)  
Row, [90](#)

**S**

ScrollColumn, [90](#), [91](#)  
Selection, [90](#)  
Shape, [90](#)  
ShapeRange, [91](#)  
Shapes, [90](#), [91](#)  
Sheet1, [90](#), [91](#)

**T**

Text, [90](#)  
TextBox, [90](#)  
TextBoxName, [90](#)  
TextFrame2, [90](#)  
TextRange, [90](#)  
Top, [91](#)

**V**

Value, [90](#), [91](#)  
Visible, [90](#), [91](#)

```

1 Option Explicit
2 Sub Empl_EventsRefresh()
3     Dim LastEventRow As Long
4     Dim LastFiltRow As Long
5     With Sheet1
6         'Load Events List Into Events Sheet Using Advanced Filter & Sort
7         Sheet7.Range("BA2:BK2,BA5:BK99999").ClearContents 'Clear any Previous Results
8         Sheet7.Range("BA2").Value = .Range("J2").Value 'Employee ID
9         LastEventRow = Sheet7.Range("A99999").End(xlUp).Row
10        On Error GoTo NoEvents
11        Sheet7.Range("A4:W" & LastEventRow).AdvancedFilter xlFilterCopy, CriteriaRange:=Sheet7
12        .Range("BA1:BA2"), CopytoRange:=Sheet7.Range("BA4:BK4"), Unique:=True
13        LastFiltRow = Sheet7.Range("BA99999").End(xlUp).Row
14        If LastFiltRow < 5 Then GoTo NoEvents
15        'Sort events from Newest To Oldest
16        Sheet7.Sort.SortFields.Clear
17        Sheet7.Sort.SortFields.Add Key:=.Range("BD5"), SortOn:=xlSortOnValues, Order:=
18        xlDescending, DataOption:=xlSortNormal
19        With Sheet7.Sort
20            .SetRange Range("BA5:BK" & LastFiltRow)
21            .Apply
22        End With
23        If LastFiltRow > 21 Then LastFiltRow = 21 'Set Maximum of 17 Rows
24        .Range("E188:H" & LastFiltRow + 183).Value = Sheet7.Range("BB5:BE" & LastFiltRow).
25        Value 'Copy First Section
26        .Range("AE188:AF188").Copy 'Copy Formulas To Combine Frequency & Start On Fields
27        .Range("I188:J" & LastFiltRow + 183).PasteSpecial xlPasteFormulas 'Paste Down
28        Formulas
29        .Range("K188:L" & LastFiltRow + 183).Value = Sheet7.Range("BJ5:BK" & LastFiltRow).
30        Value 'Copy Last Section
31        Application.CutCopyMode = xlCopy
32        NoEvents:
33    End With
34 End Sub
35
36 Sub Empl_LeaveRefresh()
37     Dim LastLeaveRow As Long
38     Dim LastFiltRow As Long
39     With Sheet1
40         Sheet11.Range("AA3,AA5:ADI99999").ClearContents 'Clear any Previous Results
41         Sheet11.Range("AA3").Value = .Range("J2").Value 'Employee ID
42         LastLeaveRow = Sheet11.Range("A99999").End(xlUp).Row
43         Sheet11.Range("A4:H" & LastLeaveRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
44         Sheet11.Range("AA2:AA3"), CopytoRange:=Sheet11.Range("AA4:AE4"), Unique:=True
45         LastFiltRow = Sheet11.Range("AA99999").End(xlUp).Row
46         If LastFiltRow < 5 Then GoTo NoLeave
47
48         If LastFiltRow > 16 Then LastFiltRow = 16 'Set Maximum of 12 Rows
49         .Range("E168:H" & LastFiltRow + 163).Value = Sheet11.Range("AA5:AD" & LastFiltRow).
50         Value 'Copy Date, Start & End Times ' Add in Leave Type, Paid, Carryover, Allowance
51         .Range("L168:L" & LastFiltRow + 163).Value = Sheet11.Range("AE5:AE" & LastFiltRow).
52         Value 'Copy Date, Start & End Times 'Add In Row #'s
53         NoLeave:
54     End With
55 End Sub
56
57 Sub Empl_Load()
58     Dim EmpRow As Long
59     Dim EmpCol As Long
60     With Sheet1
61         If .Range("B5").Value = Empty Then

```

```

54   1 2 3   MsgBox "Please enter a valid Employee from the drop down list"
55       Exit Sub
56   End If
57   StopCalc
58   .Range("B1" ).Value = True 'Set Employee Load To True
59   EmpRow = .Range("B5" ).Value
60   .Range(
61       "J2,F6:G6,I6,F8:I8,F10:G10,I10,K10,F12,H12:I12,F14,H14:I14,K14,F16,H16:I16,K16,G66,J66,G
62       68,J68,G70,J70,G73,J73,G75,J75,G86,J86,G88,J88,G182,I182,K182,E183" ).ClearContents
63   .Range(
64       "E19:K24,E28:L44,F46:G46,I46,K46,E48:L64,G93:I104,K93:K104,G108:I119,K108:K119,G128:I139
65       ,K128:K139,F148:J154,L148:L154,E168:H179,L168:HL179,E188:K204" ).ClearContents
66   For EmpCol = 1 To 33
67     .Range(Sheet2.Cells(1, EmpCol).Value).Value = Sheet2.Cells(EmpRow, EmpCol).Value
68   Next EmpCol
69   .Shapes("NewEmpGrp" ).Visible = msoFalse
70   .Shapes("ExistEmpGrp" ).Visible = msoCTrue
71   On Error Resume Next
72   .Shapes("AttachThumb" ).Delete 'Remove Attachment thumbnail, If any
73   On Error GoTo 0
74   .Range("B6" ).Value = False 'Set New Employee To False
75   .Range("E4" ).Select 'Load General Info Tab
76   If .Range("B2" ).Value = 5 Then Show_EmplPic 'Only when First Tab is selected
77   Attach_Refresh 'Refresh Attachments Table
78   Empl_TimeclockRefresh 'Refresh Time Clock Entries
79   Empl_WeeklySchedRefresh 'Refresh Weekly Schedule
80   Empl_LeaveRefresh 'Refresh Employee Leave Items
81   Empl_EventsRefresh 'Refresh Employee Events
82   Empl_PayItemsRefresh 'Refresh Payroll Tax, Other Deductions & Other Additions
83   .Range("B1" ).Value = False 'Set Employee Load To False
84   .Range("F2" ).Select
85   End With
86   ResetCalc
87   End Sub
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

```

---

```

96   Sub Empl_PayItemsRefresh()
97     Dim LastPayItemRow As Long
98     Dim LastFiltRow As Long
99     Dim PayItemRow As Long
100    Dim EmpMgrRow As Long
101    Dim ItemType As String
102    With Sheet17
103      .Range("M2:W99999" ).ClearContents 'Clear any Previous Criteria & Results
104      .Range("M2" ).Value = Sheet1.Range("J2" ).Value 'Employee ID
105      LastPayItemRow = .Range("A99999" ).End(xlUp).Row
106      .Range("A2:I" & LastPayItemRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
107          "M1:M2" ), CopytoRange:=.Range("O1:W1" ), Unique:=True
108      LastFiltRow = .Range("O99999" ).End(xlUp).Row
109      If LastFiltRow < 2 Then GoTo NoPayItem
110      For PayItemRow = 2 To LastFiltRow
111        ItemType = .Range("P" & PayItemRow).Value 'Pay Item Type
112        EmpMgrRow = .Range("V" & PayItemRow).Value 'Employee Manager Row
113        Sheet1.Range("G" & EmpMgrRow).Value = .Range("Q" & PayItemRow).Value 'Item Name
114        Sheet1.Range("K" & EmpMgrRow).Value = .Range("W" & PayItemRow).Value 'Pay Item Db
115      Next Row
116      If ItemType = "Tax Deduction" Then 'Tax Deduction Items
117        Sheet1.Range("H" & EmpMgrRow).Value = .Range("R" & PayItemRow).Value 'Wage %
118        Sheet1.Range("I" & EmpMgrRow).Value = .Range("S" & PayItemRow).Value 'Base
119      Else: ' Pay Addition or Pay Deduction
120        Sheet1.Range("H" & EmpMgrRow).Value = .Range("T" & PayItemRow).Value 'Amount

```

```

1 2 3 4
109 Sheet1.Range("I" & EmpMgrRow).Value = .Range("U" & PayItemRow).Value
      'Frequency
110 End If
111 Next PayItemRow
112 NoPayItem:
113 End With
114 End Sub

115
116 Sub Empl_TimeclockRefresh()
117 Dim LastTimeclockRow As Long
118 Dim LastFiltRow As Long
119 With Sheet1
120 .Range("E28:K44").ClearContents 'Clear Previous Results
121 Sheet9.Range("AA3:AI3,AA5:AAI99999").ClearContents 'Clear any Previous Results
122 Sheet9.Range("AB3").Value = .Range("J2").Value 'Employee ID
123 If .Range("I26").Value <> Empty Then Sheet9.Range("Z3").Value = ">=" & .Range("I26"
      ).Value Else: Sheet9.Range("Z3").ClearContents 'From Date
124 If .Range("K26").Value <> Empty Then Sheet9.Range("AD3").Value = "<=" & .Range(
      "K26").Value Else: Sheet9.Range("AD3").ClearContents 'To Date
125 LastTimeclockRow = Sheet9.Range("A99999").End(xlUp).Row
126 Sheet9.Range("A4:I" & LastTimeclockRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
      Sheet9.Range("Z2:AD3"), CopyToRange:=Sheet9.Range("AA4:AF4"), Unique:=True
127 LastFiltRow = Sheet9.Range("AA99999").End(xlUp).Row
128 If LastFiltRow < 5 Then GoTo NoTimeclock
129 'Sort TimeClock Entries Newest To Oldest
130 Sheet9.Sort.SortFields.Clear
131 Sheet9.Sort.SortFields.Add Key:=.Range("AB5"), SortOn:=xlSortOnValues, Order:=
      xlDescending, DataOption:=xlSortNormal
132 With Sheet9.Sort
133 .SetRange Range("AA5:AI" & LastFiltRow)
134 .Apply
135 End With
136 If LastFiltRow > 21 Then LastFiltRow = 21 'Set Maximum of 17 Rows
137 StopCalc
138 Sheet9.Range("AG1:AI1").Copy
139 Sheet9.Range("AG5:AI" & LastFiltRow).PasteSpecial xlPasteFormulas
140 ResetCalc
141 .Range("E28:F" & LastFiltRow + 23).Value = Sheet9.Range("AB5:AC" & LastFiltRow).Value
      'Copy Date, Start & End Times
142 .Range("G28:I" & LastFiltRow + 23).Value = Sheet9.Range("AG5:AI" & LastFiltRow).Value
      'Copy Total Break, Reg. Hours & OT Hours
143 .Range("J28:J" & LastFiltRow + 23).Value = Sheet9.Range("AF5:AF" & LastFiltRow).Value
      'Copy Notes
144 Application.CutCopyMode = xlCopy
145 NoTimeclock:
146 End With
147 End Sub

148
149
150 Sub Empl_WeeklySchedRefresh()
151 Dim WeeklySchedRow As Long
152 Dim WeeklyRow As Long
153 Dim WeeklyCol As Long
154 Dim SchedRow As Long
155 Dim EmpID As String
156 Dim WeeklySchedID As Range
157 Dim FoundWeeklySchedID As Range
158 With Sheet1
159 SchedRow = 148
160 EmpID = .Range("J2").Value
161 Set WeeklySchedID = Sheet13.Range("WeeklySched_EmplID")

```

1 2

```
162     1 2 Set FoundWeeklySchedID = WeeklySchedID.Find(EmpID, , xlValues, xlWhole)
163     If Not FoundWeeklySchedID Is Nothing Then 'ID is Found
164         WeeklyRow = FoundWeeklySchedID.Row
165         For WeeklyCol = 2 To 32 Step 5
166             .Range("F" & SchedRow & ":J" & SchedRow).Value = Range(Sheet13.Cells(WeeklyRow,
167                 WeeklyCol), Sheet13.Cells(WeeklyRow, WeeklyCol + 4)).Value
168             SchedRow = SchedRow + 1
169         Next WeeklyCol
170     End If
171
172     NoWeekly:
173 End With
174 End Sub
```

**A**

Add, 93, 95  
AdvancedFilter, 93-95  
Application, 93, 95  
Apply, 93, 95  
Attach\_Refresh, 94

**C**

Cells, 94, 96  
Clear, 93, 95  
ClearContents, 93-95  
Copy, 93, 95  
CopytoRange, 93-95  
CriteriaRange, 93-95  
CutCopyMode, 93, 95

**D**

DataOption, 93, 95  
Delete, 94

**E**

EmpCol, 93, 94  
EmpID, 95, 96  
Empl\_EventsRefresh, 93, 94  
Empl\_LeaveRefresh, 93, 94  
Empl\_Load, 93  
Empl\_PayItemsRefresh, 94  
Empl\_TimeclockRefresh, 94, 95  
Empl\_WeeklySchedRefresh, 94, 95  
EmpMgrRow, 94, 95  
EmpRow, 93, 94  
Empty, 93, 95  
Explicit, 93

**F**

Find, 96  
FoundWeeklySchedID, 95, 96

**I**

ItemType, 94

**K**

Key, 93, 95

**L**

LastEventRow, 93  
LastFiltRow, 93-95  
LastLeaveRow, 93  
LastPayItemRow, 94  
LastTimeclockRow, 95

**M**

MsgBox, 94  
msoCTrue, 94  
msoFalse, 94

**N**

NoEvents, 93  
NoLeave, 93  
NoPayItem, 94, 95  
NoTimeclock, 95  
NoWeekly, 96

**O**

Order, 93, 95

**P**

PasteSpecial, 93, 95  
PayItemRow, 94, 95

**R**

Range, 93-96  
ResetCalc, 94, 95  
Row, 93-96

**S**

SchedRow, 95, 96  
SetRange, 93, 95  
Shapes, 94  
Sheet1, 93-95  
Sheet11, 93  
Sheet13, 95, 96  
Sheet17, 94  
Sheet2, 94  
Sheet7, 93  
Sheet9, 95  
Show\_EmplPic, 94  
Sort, 93, 95  
SortFields, 93, 95  
SortOn, 93, 95  
StopCalc, 94, 95

**U**

Unique, 93-95

**V**

Value, 93-96  
Visible, 94

**W**

WeeklyCol, 95, 96  
WeeklyRow, 95, 96  
WeeklySchedID, 95, 96  
WeeklySchedRow, 95

**X**

xlCopy, 93, 95  
xlDescending, 93, 95  
xlFilterCopy, 93-95  
xlPasteFormulas, 93, 95  
xlSortNormal, 93, 95  
xlSortOnValues, 93, 95  
xlUp, 93-95  
xlValues, 96  
xlWhole, 96

```

1 Option Explicit
2
3 Sub Empl_Archive()
4 Dim EmplRow As Long
5 Dim EmplCol As Long
6 Dim ArchiveRow As Long
7 Dim EmplName As String
8 If MsgBox("Are you sure you want to Archive this employee?" & vbCrLf & "Their name will
no longer be available in the drop down list" , vbYesNo, "Archive Employee" ) = vbNo Then
Exit Sub
9 With Sheet1
10 If .Range("B4" ).Value = Empty Then
11 MsgBox "Please select an employee to Archive"
12 Exit Sub
13 End If
14 EmplName = .Range("F2" ).Value 'Employee Name
15 EmplRow = .Range("B4" ).Value 'Employee Row
16 ArchiveRow = Sheet6.Range("A99999" ).End(xlUp).Row + 1 'First Available Archive Row
17 .Range("H14" ).Value = "Inactive" 'You can change this to whatever you want
18 For EmplCol = 1 To 33
19 If Sheet2.Cells(EmplRow, EmplCol).Value <> "" Then Sheet6.Cells(ArchiveRow, EmplCol
).Value = Sheet2.Cells(EmplRow, EmplCol).Value 'Copy Over Employee Data One By one
to activate Shared Folder
20 Next EmplCol
21 Sheet2.Range("J" & EmplRow).Value = EmplName & " archived on " & Format(Sheet8.Range(
"CurrentDate" ).Value, "ddd, mmm d,yyyy" ) 'Set Archived Text
22 Sheet2.Range("B" & EmplRow).ClearContents 'Clear First Name
23 Sheet2.Range("C" & EmplRow).ClearContents 'Clear Last Name
24
25 Empl_SortByName 'Resort Employee Name List
26 Sheet1.Range("F2" ).Value = Sheet2.Range("BC4" ).Value 'Reset to display First
Employee
27 MsgBox EmplName & " has been archived"
28 End With
29 End Sub
30
31
32 Sub Empl_CancelNew()
33 Sheet1.Range("F2" ).Value = Sheet2.Range("BC4" ).Value 'Reset to display First Employee
34 End Sub
35
36 Sub Empl_New()
37 With Sheet1
38 .Range("B1" ).Value = True 'Set Employee Load To False
39 .Shapes("NewEmpGrp" ).Visible = msoCTrue
40 .Shapes("ExistEmpGrp" ).Visible = msoFalse
41 .Shapes("DefaultPicture" ).Visible = msoCTrue
42 On Error Resume Next
43 .Shapes("EmplPic" ).Delete
44 On Error GoTo 0
45 .Range(
"F2:G2,J2,F6:G6,I6,F8:I8,F10:G10,I10,K10,F12,H12:I12,F14,H14:I14,K14,F16,H16:I16,K16,G66
,J66,G68,J68,G70,J70,G73,J73,G75,J75,G86,J86,G88,J88,G182,I182,K182,E183" ).
ClearContents
46 .Range(
"F26:G26,I26,K26,E19:K24,E28:L44,F46:G46,I46,K46,E48:L64,G93:I104,K93:K104,G108:I119,K10
8:K119,G128:I139,K128:K139,F148:J154,L148:L154,E168:H179,L168:HL179,E188:K204" ).
ClearContents
47
48 'Add New Employee Default Values
49 .Range("G66" ).Value = Sheet4.Range("E108" ).Value 'Default Pay Type

```



```

1 2
50 .Range("J66" ).Value = Sheet4.Range("E109" ).Value 'Default Salary Frequency
51 .Range("G68" ).Value = Sheet4.Range("E110" ).Value 'Default Hourly Amount
52 .Range("J68" ).Value = Sheet4.Range("E111" ).Value 'Default OT Amount
53 .Range("G70" ).Value = Sheet4.Range("E112" ).Value 'Default Hours Per Week
54 .Range("J70" ).Value = Sheet4.Range("E113" ).Value 'Default Weeks Per Year
55 .Range("G73" ).Value = Sheet4.Range("E114" ).Value 'Default Custom Field 1
56 .Range("J73" ).Value = Sheet4.Range("E115" ).Value 'Default Custom Field 2
57 .Range("G75" ).Value = Sheet4.Range("E116" ).Value 'Default Custom Field 3
58 .Range("J75" ).Value = Sheet4.Range("E117" ).Value 'Default Custom Field 4
59 .Range("G108:I119" ).Value = Sheet4.Range("D129:F140" ).Value 'Default Payroll
    Deductions
60 .Range("G128:I139" ).Value = Sheet4.Range("H129:J140" ).Value 'Default Payroll
    Additions
61 .Range("J2" ).Value = Application.WorksheetFunction.Max(Sheet2.Range("EmployeeID" )) +
    1 'First Available Employee ID
62 .Range("B6" ).Value = True 'Set New Employee To True
63 .Range("E4" ).Select 'Selects General Info Tab
64 .Range("F6" ).Select 'Select Last Name Field
65
66 .Range("B1" ).Value = False 'Set Employee Load To False
67
68     End With
69 End Sub
70
71 Sub Empl_ReActivate()
72     Dim EmpRow As Long
73     Dim EmpCol As Long
74     Dim ArchiveRow As Long
75     Dim EmpName As String
76     Dim EmpID As String
77     Dim EmpIDRng As Range
78     Dim FoundEmpIDRng As Range
79     If MsgBox("Are you sure you want to reactivate this Employee?" , vbYesNo, "Reactivate
    Employee" ) = vbNo Then Exit Sub
80     With Sheet6
81         If .Range("BA1" ).Value = Empty Then
82             MsgBox "Please select an Employee to Reactivate"
83             Exit Sub
84         End If
85         ArchiveRow = .Range("BA1" ).Value 'Employee Row
86         .Range("J" & ArchiveRow).Value = "Active" 'You can change this to whatever you want
87
88         Set EmpIDRng = Sheet2.Range("EmployeeID" )
89         EmpID = Sheet6.Range("A" & ArchiveRow).Value 'Employee ID
90         Set FoundEmpIDRng = EmpIDRng.Find(EmpID, , xlValues, xlWhole)
91         If Not FoundEmpIDRng Is Nothing Then
92             EmpRow = FoundEmpIDRng.Row
93         Else: Exit Sub
94         End If
95         For EmpCol = 2 To 33
96             If Sheet6.Cells(ArchiveRow, EmpCol).Value <> "" Then
97                 Sheet2.Cells(EmpRow, EmpCol).Value = Sheet6.Cells(ArchiveRow, EmpCol).Value
98                 Sheet6.Cells(ArchiveRow, EmpCol).ClearContents 'Clear Archive
99             End If
100        Next EmpCol
101
102        EmpName = Sheet2.Range("BC" & EmpRow).Value 'Employee Name
103        Sheet6.Range("J" & ArchiveRow).Value = EmpName & " unarchived on " & Format(Sheet8.
    Range("CurrentDate" ).Value, "ddd, mmm d,yyyy" ) 'Set Unarchived Text
104        Empl_SortByName 'Resort List

```

1 2

```

1 2
105     MsgBox EmpName & " has been reactivated"
106     End With
107     End Sub
108
109
110
111 Sub Empl_SaveNew()
112     Dim EmpRow As Long
113     Dim EmpCol As Long
114     With Sheet1
115         If .Range("F6").Value = Empty Then
116             MsgBox "Please enter a Last Name for the Employee"
117             .Range("F6").Select
118             Exit Sub
119         End If
120         EmpRow = Sheet2.Range("A99999").End(xlUp).Row + 1 'First Avail Row
121         Sheet2.Range("A" & EmpRow).Value = Application.WorksheetFunction.Max(Sheet2.Range(
122             "EmployeeID" )) + 1
123         For EmpCol = 2 To 33
124             If .Range(Sheet2.Cells(1, EmpCol).Value).Value <> Empty Then Sheet2.Cells(EmpRow,
125                 EmpCol).Value = .Range(Sheet2.Cells(1, EmpCol).Value).Value 'update only if not
126                 blank
127             Next EmpCol
128             .Range("F2").Value = .Range("F6").Value & ", " & .Range("I6").Value 'Add in Last
129             Name, First Name
130             .Shapes("NewEmpGrp").Visible = msoFalse
131             .Shapes("ExistEmpGrp").Visible = msoTrue
132             .Range("B1").Value = False 'Set Empl Load To False
133             .Range("B6").Value = False 'Set New Employee To False
134         End With
135         Empl_SortByName 'Sort Employee List
136         Sheet8.Range("D8:D314").Formula = Sheet8.Range("D8").Formula 'Update Employee Schedule
137         Names
138     End Sub
139
140
141
142 Sub Empl_SortByName()
143     'Sort Employee List
144     StopCalc
145     Dim LastEmpRow As Long
146     With Sheet2
147         LastEmpRow = .Range("A99999").End(xlUp).Row 'Last Row of value
148         .Range("BB4:BB" & LastEmpRow).Value = .Range("BC4:BC" & LastEmpRow).Value 'Copy Over
149         Names
150         .Sort.SortFields.Clear
151         .Sort.SortFields.Add Key:=.Range("BB4"), SortOn:=xlSortOnValues, Order:=xlAscending,
152         DataOption:=xlSortNormal
153         With .Sort
154             .SetRange Range("BB4:BB" & LastEmpRow)
155             .Apply
156         End With
157     End With
158     ResetCalc
159 End Sub

```

**A**

Add, 100  
Application, 99, 100  
Apply, 100  
ArchiveRow, 98, 99

**C**

Cells, 98-100  
Clear, 100  
ClearContents, 98, 99

**D**

DataOption, 100  
Delete, 98

**E**

EmpCol, 99, 100  
EmpID, 99  
EmpIDRng, 99  
Empl\_Archive, 98  
Empl\_CancelNew, 98  
Empl\_New, 98  
Empl\_ReActivate, 99  
Empl\_SaveNew, 100  
Empl\_SortByName, 98-100  
EmplCol, 98  
EmplName, 98  
EmplRow, 98  
EmpName, 99, 100  
EmpRow, 99, 100  
Empty, 98-100  
Explicit, 98

**F**

Find, 99  
Format, 98, 99  
Formula, 100  
FoundEmpIDRng, 99

**K**

Key, 100

**L**

LastEmpRow, 100

**M**

Max, 99, 100  
MsgBox, 98-100  
msoCTrue, 98, 100  
msoFalse, 98, 100

**O**

Order, 100

**R**

Range, 98-100  
ResetCalc, 100  
Row, 98-100

**S**

SetRange, 100  
Shapes, 98, 100  
Sheet1, 98, 100  
Sheet2, 98-100  
Sheet4, 98, 99  
Sheet6, 98, 99  
Sheet8, 98-100  
Sort, 100  
SortFields, 100

SortOn, 100  
StopCalc, 100

**V**

Value, 98-100  
vbCrLf, 98  
vbNo, 98, 99  
vbYesNo, 98, 99  
Visible, 98, 100

**W**

WorksheetFunction, 99, 100

**X**

xlAscending, 100  
xlSortNormal, 100  
xlSortOnValues, 100  
xlUp, 98, 100  
xlValues, 99  
xlWhole, 99

```

1  Option Explicit
2
3  Sub Add_EmplPic()
4      Dim fso As Object
5      Dim PicFile As FileDialog
6      Dim FileName As String
7      Dim OrigFilePath As String
8      Dim DestFileFolder As String
9      Dim DestFilePath As String
10     Dim MainAppFolder As String
11     Dim EmpID As String
12     Set fso = CreateObject("Scripting.FileSystemObject") 'Set File System Object
13     CheckMainAppFolder 'Run Macro to Check For Correct Main Application Folder
14     If Dir(Sheet4.Range("ApplicationFolder").Value, vbDirectory) = "" Then Exit Sub 'Exit
15     If folder still does not exist
16     MainAppFolder = Sheet4.Range("ApplicationFolder").Value 'Set Main Application Folder
17
18     'Check For Attachments Folder
19     If Dir(MainAppFolder & "/Attachments", vbDirectory) = "" Then 'If attach folder does
20     not exist, Create Folder & Exit Sub
21     fso.createFolder (MainAppFolder & "/Attachments") 'Add Folder if needed
22     End If
23
24     With Sheet1
25     If .Range("B6").Value = True Then Empl_SaveNew 'Save Employee Before moving forward
26     to lock in Employee ID
27     EmpID = .Range("J2").Value 'Set Employee ID
28     Set PicFile = Application.FileDialog(msoFileDialogFilePicker) 'Open File Dialog
29     With PicFile
30     .Title = "Select an Employee Picture"
31     .Filters.Add "All Picture Files", "*.jpg,*jpeg,*gif*.png*.gif*.bmp*.tiff"
32     If .Show <> -1 Then GoTo NoSelection
33     OrigFilePath = .SelectedItems(1)
34     End With
35
36     DestFileFolder = MainAppFolder & "\Attachments\" & EmpID & "_Emp_ID" 'Get Main
37     Folder for Employee pictuers and attachments
38     If Dir(DestFileFolder, vbDirectory) = "" Then 'If Employee Folder does not Exist,
39     then create it
40     fso.createFolder (DestFileFolder) 'Add Folder if needed
41     End If
42
43     FileName = Dir(OrigFilePath)
44     'Check if Employee Folder exists, if not, add it.
45     DestFilePath = DestFileFolder & "\" & FileName
46     FileCopy OrigFilePath, DestFilePath 'Copy user selected file into appropriate
47     Employee folder
48     'Kill (OrigFilePath) 'If you want to delete the original file in the original location
49
50     Sheet1.Range("K10").Value = Dir(DestFilePath) 'Put Only File Name in K10 (Path is
51     Dynamic based on user)
52     Show_EmplPic
53     NoSelection:
54     End With
55 End Sub
56
57 Sub Show_EmplPic()
58     Dim PicFileName As String
59     Dim EmpID As String
60     Dim MainAppFolder As String

```

```
1
54 Dim LongFileName As String
55 With Sheet1
56     On Error Resume Next
57     .Shapes("EmplPic" ).Delete 'Delete Picture if it exists
58     On Error GoTo 0
59     MainAppFolder = Sheet4.Range("ApplicationFolder" ).Value 'Set Main Application Folder
60     If MainAppFolder = "" Then GoTo DefaultPic 'If no main application folder then use
61     If .Range("B6" ).Value = True Then GoTo DefaultPic 'If new Employee GoTo Default
62     picture
63     EmpID = .Range("J2" ).Value 'Set Employee ID
64     PicFileName = .Range("K10" ).Value 'Path of the picture
65     If PicFileName = Empty Then
66         .Shapes("DefaultPicture" ).Visible = msoCTrue
67         Exit Sub
68     End If
69     LongFileName = MainAppFolder & "\Attachments\" & EmpID & "_Emp_ID\" & PicFileName
70     .Shapes("DefaultPicture" ).Visible = msoFalse
71     On Error GoTo DefaultPic
72     With .Pictures.Insert(LongFileName)
73         With .ShapeRange
74             .LockAspectRatio = msoTrue
75             .Height = 95
76             .Name = "EmplPic"
77         End With 'Shape Range
78     End With 'Pictures
79
80     With .Shapes("EmplPic" )
81         .Left = Sheet1.Range("J5" ).Left
82         .Top = Sheet1.Range("J5" ).Top
83         .IncrementLeft 40
84         .IncrementTop 10
85     End With
86     Exit Sub:
87     DefaultPic:
88     .Shapes("DefaultPicture" ).Visible = msoCTrue
89 End With 'Sheet 1
90
91 End Sub
```

**A**

Add, [102](#)  
Add\_EmplPic, [102](#)  
Application, [102](#)

**C**

CheckMainAppFolder, [102](#)  
createFolder, [102](#)  
CreateObject, [102](#)

**D**

DefaultPic, [103](#)  
Delete, [103](#)  
DestFileFolder, [102](#)  
DestFilePath, [102](#)  
Dir, [102](#)

**E**

EmpID, [102](#), [103](#)  
Empl\_SaveNew, [102](#)  
Empty, [103](#)  
Explicit, [102](#)

**F**

FileCopy, [102](#)  
FileDialog, [102](#)  
FileName, [102](#)  
Filters, [102](#)  
fso, [102](#)

**H**

Height, [103](#)

**I**

IncrementLeft, [103](#)  
IncrementTop, [103](#)  
Insert, [103](#)

**L**

Left, [103](#)  
LockAspectRatio, [103](#)  
LongFileName, [103](#)

**M**

MainAppFolder, [102](#), [103](#)  
msoCTrue, [103](#)  
msoFalse, [103](#)  
msoFileDialogFilePicker, [102](#)  
msoTrue, [103](#)

**N**

Name, [103](#)  
NoSelection, [102](#)

**O**

OrigFilePath, [102](#)

**P**

PicFile, [102](#)  
PicFileName, [102](#), [103](#)  
Pictures, [103](#)

**R**

Range, [102](#), [103](#)

**S**

SelectedItems, [102](#)  
ShapeRange, [103](#)  
Shapes, [103](#)

Sheet1, [102](#), [103](#)  
Sheet4, [102](#), [103](#)  
Show, [102](#)  
Show\_EmplPic, [102](#)

**T**

Title, [102](#)  
Top, [103](#)

**V**

Value, [102](#), [103](#)  
vbDirectory, [102](#)  
Visible, [103](#)

```
1 Option Explicit
2
3
4 Sub Empl_SwitchHorizontalTabs()
5     Dim SelCol As Long
6     Dim FirstRow As Long
7     StopCalc
8     With Sheet1
9         SelCol = .Range("B2" ).Value
10
11         If SelCol > 8 Then SelCol = SelCol + 3
12         .Range("5:204" ).EntireRow.Hidden = True
13         FirstRow = 5 + ((SelCol - 5) * 20)
14         .Range(FirstRow & ":" & FirstRow + 19).EntireRow.Hidden = False
15     If SelCol <> 5 Then
16         On Error Resume Next
17         .Shapes("GenInfoGrp" ).Visible = msoFalse
18         On Error GoTo 0
19         .Shapes("DefaultPicture" ).Visible = msoFalse
20         On Error Resume Next
21         .Shapes("EmplPic" ).Visible = msoFalse
22         On Error GoTo 0
23         On Error Resume Next
24         .Shapes("AttachThumb" ).Visible = msoFalse
25         On Error GoTo 0
26     Else:
27         On Error Resume Next
28         .Shapes("GenInfoGrp" ).Visible = msoCTrue
29         On Error GoTo 0
30         If .Range("K10" ).Value = Empty Then .Shapes("DefaultPicture" ).Visible = msoCTrue
31         Else: Show_EmplPic
32     End If
33     If SelCol = 8 Then .Range("B3" ).Value = 66
34     'Select First Cell in Tab
35     If .Range("B2" ).Value = 5 Then .Range("F6" ).Select
36     If .Range("B2" ).Value = 6 Then .Range("F26" ).Select
37     If .Range("B2" ).Value = 7 Then .Range("F46" ).Select
38     If .Range("B2" ).Value = 8 Then .Range("G66" ).Select
39     If .Range("B2" ).Value = 9 Then .Range("F148" ).Select
40     If .Range("B2" ).Value = 10 Then .Range("E168" ).Select
41     If .Range("B2" ).Value = 11 Then .Range("E188" ).Select
42
43 End With
44 ResetCalc
45 End Sub
46
47 Sub EmpTab_Events()
48     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("K4" ).Select
49 End Sub
50
51 Sub EmpTab_GenInfo()
52     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("E4" ).Select
53 End Sub
54
55 Sub EmpTab_Leave()
56     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("J4" ).Select
57 End Sub
58
59 Sub EmpTab_PayDetails()
60     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("H4" ).Select
61 End Sub
62
63 Sub EmpTab_PayHistory()
64     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("G4" ).Select
65 End Sub
66
67 Sub EmpTab_Scheduling()
```

1

```
1
61   If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("I4" ).Select
62   End Sub
63   Sub EmpTab_TimeClock()
64   If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("F4" ).Select
65   End Sub
66
67
68   Sub SwitchVerticalTabs()
69   Dim SelRow As Long
70   Dim FirstRow As Long
71   SelRow = Right(ActiveCell.Row, 1) - 5
72   With Sheet1
73   .Range("65:144" ).EntireRow.Hidden = True
74   FirstRow = 65 + ((SelRow - 1) * 20)
75   .Range(FirstRow & ":" & FirstRow + 19).EntireRow.Hidden = False
76   .Range("B3" ).Value = SelRow + FirstRow
77   .Range("F2" ).Select
78   End With
79   End Sub
```



## A

ActiveCell, [106](#)  
ActiveSheet, [105](#), [106](#)

## C

CodeName, [105](#), [106](#)

## E

Empl\_SwitchHorizontalTabs, [105](#)  
EmpTab\_Events, [105](#)  
EmpTab\_GenInfo, [105](#)  
EmpTab\_Leave, [105](#)  
EmpTab\_PayDetails, [105](#)  
EmpTab\_PayHistory, [105](#)  
EmpTab\_Scheduling, [105](#)  
EmpTab\_TimeClock, [106](#)  
Empty, [105](#)  
EntireRow, [105](#), [106](#)  
Explicit, [105](#)

## F

FirstRow, [105](#), [106](#)

## H

Hidden, [105](#), [106](#)

## M

msoCTrue, [105](#)  
msoFalse, [105](#)

## R

Range, [105](#), [106](#)  
ResetCalc, [105](#)  
Right, [106](#)  
Row, [106](#)

## S

SelCol, [105](#)  
SelRow, [106](#)  
Shapes, [105](#)  
Sheet1, [105](#), [106](#)  
Show\_EmplPic, [105](#)  
StopCalc, [105](#)  
SwitchVerticalTabs, [106](#)

## V

Value, [105](#), [106](#)  
Visible, [105](#)

```
1 Option Explicit
2
3 Sub Events_ClearFilter()
4     With Sheet5
5         Sheet7.Range("AD2:AQ2" ).ClearContents
6         .Range("D26" ).Value = "Enter Name"
7         .Range("E26" ).Value = "Enter Type"
8         .Range("G26" ).Value = "Enter Employee"
9         Event_Refresh 'Run Events Refresh Macro
10    End With
11 End Sub
12
13 Sub Events_FilterTable()
14     With Sheet5
15         Sheet7.Range("AD2:AQ2,AD5:AQ99999" ).ClearContents 'Clear prior criteria results
16         If InStr(.Range("D26" ).Value, "Enter" ) = 0 Then Sheet7.Range("AD2" ).Value = "*" & .
17         Range("D26" ).Value & "*" 'Event Name Filter
18         If InStr(.Range("E26" ).Value, "Enter" ) = 0 Then Sheet7.Range("AE2" ).Value = "*" & .
19         Range("E26" ).Value & "*" 'Event Type Filter
20         If InStr(.Range("G26" ).Value, "Enter" ) = 0 Then Sheet7.Range("AG2" ).Value = "*" & .
21         Range("G26" ).Value & "*" 'Employee Name
22         Event_Refresh
23     End With
24 End Sub
```

**C**

ClearContents, [108](#)

**E**

Event\_Refresh, [108](#)

Events\_ClearFilter, [108](#)

Events\_FilterTable, [108](#)

Explicit, [108](#)

**I**

InStr, [108](#)

**R**

Range, [108](#)

**S**

Sheet5, [108](#)

Sheet7, [108](#)

**V**

Value, [108](#)

```
1 Option Explicit
2 Dim EventRow As Long
3 Dim EventCol As Long
4 Dim MapRng As String
5
6 Sub DeleteFilters()
7     On Error Resume Next
8     Sheet7.Names("Criteria" ).Delete
9     Sheet7.Names("Extract" ).Delete
10    On Error GoTo 0
11 End Sub
12 Sub Event_AddEmailAttach()
13     Dim FileFldr As FileDialog
14     Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
15     With FileFldr
16         .Title = "Select file to attach"
17         .Filters.Add "All Files" , "*.*"
18         If .Show <> -1 Then GoTo NoSelection
19         Sheet5.Range("F16" ).Value = .SelectedItems(1)
20     End With
21     NoSelection:
22 End Sub
23 Sub Event_AddNew()
24     With Sheet5
25         .Range("B1" ).Value = True 'Set Event Load To True
26         .Range("B2" ).Value = True 'Set New Event To True
27         .Range(
28             "B4,B6,E4,H4,K4,E6:N8,F10,G10,J10,L10,N10,J11,L11,F13,G13,N13,F15:I15,K15:M15,F16:M16,F1
29             7:N17,F18:N21" ).ClearContents
30         .Shapes("NewEventGrp" ).Visible = msoCTrue
31         .Shapes("ExistEventGrp" ).Visible = msoFalse
32         .Range("N4" ).Value = Now 'Set Current Time & Date
33         .Range("B1" ).Value = False 'Set Event Load To False
34         .Range("E4" ).Select
35     End With
36 End Sub
37 Sub Event_AddNotifDefaults()
38     'If Email Notification Default Exists, Load Default Details
39     With Sheet5
40         If .Range("B18" ).Value <> Empty Then 'Default Notification Exists
41             Dim NotifRow As Long
42             Dim NotifCol As Long
43             NotifRow = .Range("B18" ).Value 'Notificaiton row
44             For NotifCol = 5 To 9
45                 .Range(Sheet4.Cells(543, NotifCol).Value).Value = Sheet4.Cells(NotifRow, NotifCol
46                 ).Value 'Add In Notification Defaults
47             Next NotifCol
48         End If
49     End With
50 End Sub
51 Sub Event_AllDay()
52     Sheet5.Range("L10" ).Value = Sheet4.Range("J8" ).Value 'Start Time
53     Sheet5.Range("L11" ).Value = Sheet4.Range("J9" ).Value 'End Time
54 End Sub
55 Sub Event_CancelNew()
56     Sheet5.Range("D27" ).Select
57 End Sub
58 Sub Event_Delete()
59     If MsgBox("Are you sure you want to delete this event?" , vbYesNo, "Delete Event" ) = vbNo
```

1

```

1
Then Exit Sub
59 With Sheet5
60   If .Range("B3").Value = Empty Then Exit Sub
61   EventRow = .Range("B3").Value 'Event Row
62   Sheet7.Range(EventRow & ":" & EventRow).EntireRow.Delete
63   Event_Refresh 'Refresh Events
64   .Range("E27").Select
65 End With
66 End Sub
67
68
69 Sub Event_Load()
70   Dim SelRow As Long
71   With Sheet5
72     SelRow = .Range("B10").Value
73     If .Range("B3").Value = Empty Then
74       MsgBox "Please select on an Event to display the Event details"
75       Exit Sub
76     End If
77     StopCalc
78     .Range("B1").Value = True 'Set Event Load To True
79     EventRow = .Range("B3").Value
80
81     For EventCol = 1 To 26
82       MapRng = Sheet7.Cells(1, EventCol).Value
83       If MapRng = "D10" Or MapRng = "D13" Then 'For Recurring & Reminders
84         If MapRng = "D10" Then
85           If Sheet7.Cells(EventRow, EventCol).Value = "Yes" Then .Range(MapRng).Value =
86             "p Recurring" Else: .Range(MapRng).Value = "o Recurring"
87           .Range(MapRng).Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
88           .Range(MapRng).Characters(Start:=2, Length:=16).Font.Name = "Calibri"
89         End If
90         If MapRng = "D13" Then
91           If Sheet7.Cells(EventRow, EventCol).Value = "Yes" Then .Range(MapRng).Value =
92             "p Reminders" Else: .Range(MapRng).Value = "o Reminders"
93           .Range(MapRng).Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
94           .Range(MapRng).Characters(Start:=2, Length:=15).Font.Name = "Calibri"
95         End If
96       Else: 'For All other fields
97         .Range(MapRng).Value = Sheet7.Cells(EventRow, EventCol).Value 'Add Mapped Values
98       End If
99     Next EventCol
100   If .Range("D13").Value = "p Reminders" Then .Shapes("EventRemindIcons").Visible =
101     msoCTrue Else .Shapes("EventRemindIcons").Visible = msoFalse 'Show Hide Reminder
102   Icon Group
103   .Shapes("NewEventGrp").Visible = msoFalse
104   .Shapes("ExistEventGrp").Visible = msoTrue
105   If .Range("H4").Value = "Leave" Or .Range("H4").Value = "Training" Then .Shapes(
106     "AllDayBtn").Visible = msoCTrue Else: .Shapes("AllDayBtn").Visible = msoFalse
107   .Range("B2").Value = False 'Set New Event To False
108   .Range("B1").Value = False 'Set Event Load To False
109   ResetCalc
110 End With
111 End Sub
112
113 Sub Event_Refresh()
114   Dim LastEventRow As Long
115   Dim LastFiltRow As Long
116   With Sheet5
117     StopCalc
118
119 1 2

```

```

1 2
113 'Load Events List Into Events Sheet Using Advanced Filter & Sort
114 .Range("D27:O99999" ).ClearContents 'Clear Existing Events list
115 Sheet7.Range("AD5:AQ99999" ).ClearContents 'Clear any Previous Results
116 LastEventRow = Sheet7.Range("A99999" ).End(xlUp).Row
117 DeleteFilters
118
119 Sheet7.Range("A4:AA" & LastEventRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
Sheet7.Range("AD1:AQ2" ), CopytoRange:=Sheet7.Range("AD4:AQ4" ), Unique:=True
120 LastFiltRow = Sheet7.Range("AD99999" ).End(xlUp).Row
121 If LastFiltRow < 5 Then GoTo NoEvent
122 'Sort Events From Newest To Oldest
123 Sheet7.Sort.SortFields.Clear
124 Sheet7.Sort.SortFields.Add Key:=.Range("AF5" ), SortOn:=xlSortOnValues, Order:=
xlDescending, DataOption:=xlSortNormal
125 With Sheet7.Sort
126 .SetRange Range("AD5:AQ" & LastFiltRow)
127 .Apply
128 End With
129 .Range("D27:I" & LastFiltRow + 22).Value = Sheet7.Range("AD5:AI" & LastFiltRow).Value
'Copy First Section
130 .Range("A27:B27" ).Copy 'Copy Formulas To Combine Frequency & Start On Fields
131 .Range("J27:K" & LastFiltRow + 22).PasteSpecial xlPasteFormulas 'Paste Down Formulas
132 .Range("L27:O" & LastFiltRow + 22).Value = Sheet7.Range("AN5:AQ" & LastFiltRow).Value
'Copy Second Section
133 Application.CutCopyMode = xlCopy
134 .Range("D27" ).Select
135 NoEvent:
136 ResetCalc
137 End With
138 End Sub
139 Sub Event_SaveNew()
140 With Sheet7
141 'Check for Required Fields
142 If Sheet5.Range("E4" ).Value = Empty Or Sheet5.Range("H4" ).Value = Empty Then
143 MsgBox "Event Name and Event Type are required fields for Events"
144 Exit Sub
145 End If
146 EventRow = .Range("A99999" ).End(xlUp).Row + 1 'First Available Event List Row
147 .Cells(EventRow, 1).Value = Sheet5.Range("B7" ).Value 'New Event ID
148 For EventCol = 2 To 23
149 If EventCol = 8 Or EventCol = 15 Then 'Reminders & Recurring Options
150 If InStr(Sheet5.Range(.Cells(1, EventCol).Value).Value, "p" ) > 0 Then .Cells(
EventRow, EventCol).Value = "Yes" Else: .Cells(EventRow, EventCol).Value = "No"
151 Else:
152 .Cells(EventRow, EventCol).Value = Sheet5.Range(.Cells(1, EventCol).Value).Value
'Event Field
153 End If
154 Next EventCol
155 .Range("AB" & EventRow).Formula = "=Row()"
156 .Range(EventRow & ":" & EventRow).WrapText = False
157 End With
158 With Sheet5
159 .Shapes("NewEventGrp" ).Visible = msoFalse
160 .Shapes("ExistEventGrp" ).Visible = msoTrue
161 .Range("B2" ).Value = False 'Set New Event To False
162 .Range("B4" ).Value = Sheet7.Cells(EventRow, 1).Value 'New Event ID
163 Event_Refresh ' Reload Events
164 End With
165 End Sub

```

**A**

Add, [110](#), [112](#)  
AdvancedFilter, [112](#)  
Application, [110](#), [112](#)  
Apply, [112](#)

**C**

Cells, [110-112](#)  
Characters, [111](#)  
Clear, [112](#)  
ClearContents, [110](#), [112](#)  
Copy, [112](#)  
CopytoRange, [112](#)  
CriteriaRange, [112](#)  
CutCopyMode, [112](#)

**D**

DataOption, [112](#)  
Delete, [110](#), [111](#)  
DeleteFilters, [110](#), [112](#)

**E**

Empty, [110-112](#)  
EntireRow, [111](#)  
Event\_AddEmailAttach, [110](#)  
Event\_AddNew, [110](#)  
Event\_AddNotifDefaults, [110](#)  
Event\_AllDay, [110](#)  
Event\_CancelNew, [110](#)  
Event\_Delete, [110](#)  
Event\_Load, [111](#)  
Event\_Refresh, [111](#), [112](#)  
Event\_SaveNew, [112](#)  
EventCol, [110-112](#)  
EventRow, [110-112](#)  
Explicit, [110](#)

**F**

FileDialog, [110](#)  
FileFldr, [110](#)  
Filters, [110](#)  
Font, [111](#)  
Formula, [112](#)

**I**

InStr, [112](#)

**K**

Key, [112](#)

**L**

LastEventRow, [111](#), [112](#)  
LastFiltRow, [111](#), [112](#)  
Length, [111](#)

**M**

MapRng, [110](#), [111](#)  
MsgBox, [110-112](#)  
msoCTrue, [110](#), [111](#)  
msoFalse, [110-112](#)  
msoFileDialogFilePicker, [110](#)  
msoTrue, [111](#), [112](#)

**N**

Name, [111](#)  
Names, [110](#)  
NoEvent, [112](#)  
NoSelection, [110](#)  
NotifCol, [110](#)

NotifRow, [110](#)  
Now, [110](#)

**O**

Order, [112](#)

**P**

PasteSpecial, [112](#)

**R**

Range, [110-112](#)  
ResetCalc, [111](#), [112](#)  
Row, [112](#)

**S**

SelectedItems, [110](#)  
SelRow, [111](#)  
SetRange, [112](#)  
Shapes, [110-112](#)  
Sheet4, [110](#)  
Sheet5, [110-112](#)  
Sheet7, [110-112](#)  
Show, [110](#)  
Sort, [112](#)  
SortFields, [112](#)  
SortOn, [112](#)  
Start, [111](#)  
StopCalc, [111](#)

**T**

Title, [110](#)

**U**

Unique, [112](#)

**V**

Value, [110-112](#)  
vbNo, [110](#)  
vbYesNo, [110](#)  
Visible, [110-112](#)

**W**

WrapText, [112](#)

**X**

xlCopy, [112](#)  
xlDescending, [112](#)  
xlFilterCopy, [112](#)  
xlPasteFormulas, [112](#)  
xlSortNormal, [112](#)  
xlSortOnValues, [112](#)  
xlUp, [112](#)

```
1  Option Explicit
2
3
4  Sub EmployIDNotFound()
5      Dim i As Integer
6      Dim delay As Double
7      Dim StartTime As Double
8      With ActiveSheet.Shapes("IDNotFound" )
9          .Visible = True
10         For i = 1 To 150
11             .Fill.Transparency = i / 150
12             delay = 0.009
13             StartTime = Timer
14             Do
15                 DoEvents
16                 Loop While Timer - StartTime < delay
17             Next i
18         .Visible = False
19     End With
20 End Sub
21
22
```



**A**

ActiveSheet, [114](#)

**D**

delay, [114](#)

DoEvents, [114](#)

**E**

EmployIDNotFound, [114](#)

Explicit, [114](#)

**F**

Fill, [114](#)

**I**

i, [114](#)

**S**

Shapes, [114](#)

StartTime, [114](#)

**T**

Timer, [114](#)

Transparency, [114](#)

**V**

Visible, [114](#)

```
1 Option Explicit
2 Sub GetFolder()
3     Dim PDFFldr As FileDialog
4     Dim sItem As String
5     Dim strPath As String
6     Set PDFFldr = Application.FileDialog(msoFileDialogFolderPicker)
7     With PDFFldr
8         .Title = "Select a Folder"
9         .AllowMultiSelect = False
10        .InitialFileName = strPath
11        If .Show <> -1 Then GoTo NextCode
12        sItem = .SelectedItems(1)
13    End With
14    Sheet1.Range("S3" ).Value = sItem
15    NextCode:
16    'GetFolder = sItem
17    Set PDFFldr = Nothing
18 End Sub
19
20 Sub MailToSelected()
21     Dim SelRow As Long
22     Dim BeregnerNm As String
23     Dim EmailAdd As String
24     Dim BeregnerRng As Range
25     Dim FoundBeregner As Range
26     SelRow = ActiveCell.Row
27     BeregnerNm = Sheet1.Range("D" & SelRow).Value
28     Set BeregnerRng = Sheet2.Range("04:09999" ) 'Not sure what row this starts on
29     Set FoundBeregner = BeregnerRng.Find(BeregnerNm, , xlValues, xlWhole)
30     If Not FoundBeregner Is Nothing Then EmailAdd = Sheet2.Range("P" & FoundBeregner.Row).
31     Value
32     'Add the rest of your email code here
33 End Sub
34
35
36
```

## A

ActiveCell, [116](#)  
AllowMultiSelect, [116](#)  
Application, [116](#)

## B

BeregnerNm, [116](#)  
BeregnerRng, [116](#)

## E

EmailAdd, [116](#)  
Explicit, [116](#)

## F

FileDialog, [116](#)  
Find, [116](#)  
FoundBeregner, [116](#)

## G

GetFolder, [116](#)

## I

InitialFileName, [116](#)

## M

MailToSelected, [116](#)  
msoFileDialogFolderPicker, [116](#)

## N

NextCode, [116](#)

## P

PDFFldr, [116](#)

## R

Range, [116](#)  
Row, [116](#)

## S

SelectedItems, [116](#)  
SelRow, [116](#)  
Sheet1, [116](#)  
Sheet2, [116](#)  
Show, [116](#)  
sItem, [116](#)  
strPath, [116](#)

## T

Title, [116](#)

## V

Value, [116](#)

## X

xlValues, [116](#)  
xlWhole, [116](#)

```

1  Option Explicit
2
3
4  Sub Notification_Close()
5      ActiveSheet.Shapes("NotificationGrp").Delete
6  End Sub
7
8  Sub Notifications_Check()
9      Dim EventRow As Long
10     Dim LastRow As Long
11     Dim LastFiltRow
12     With Sheet7
13         .Range("CA2").Value = "Yes"
14         .Range("CD2").Value = "<" & Date + 1
15         .Range("CE2").Value = "="
16         .Range("CA5:CW9999").ClearContents
17         LastRow = .Range("A99999").End(xlUp).Row
18         .Range("A4:AB" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
19             "CA1:CE2"), CopytoRange:=.Range("CA4:CW4"), Unique:=True
20         LastFiltRow = .Range("CA9999").End(xlUp).Row 'Get Last Results Row
21         If LastFiltRow < 5 Then GoTo NoNotif
22
23         'Sort List Based on Set Reminder For Date & Time (Oldest to Newest)
24         .Sort.SortFields.Clear 'Clear Any Existing Sorts
25         .Sort.SortFields.Add Key:=.Range("CV5"), SortOn:=xlSortOnValues, Order:=xlAscending,
26             DataOption:=xlSortNormal
27     With .Sort
28         .SetRange Range("CA5:CW" & LastFiltRow)
29         .Apply
30     End With
31     If .Range("CV5").Value < Now Then 'Send Now
32         Notifications_Create
33     Else: 'Schedule Send
34         ' Application.OnTime .Range("CV5").Value, "Notifications_Create"
35     End If
36     NoNotif:
37 End With
38 End Sub
39
40 Sub Notifications_Create()
41     Dim OutApp As Object
42     Dim OutMail As Object
43     Dim EventName As String
44     Dim EventType As String
45     Dim EventNotes As String
46     Dim EventCreatedOn As Date
47     Dim StartOn As Date
48     Dim EndOn As Date
49     Dim EmployeeName As String
50     Dim EmployeeEmail As String
51     Dim EmailTo As String
52     Dim EmailBCC As String
53     Dim UserName As String
54     Dim UserEmail As String
55     Dim LeaveType As String
56     Dim EmplID As String
57     Dim Subj As String
58     Dim Mess As String
59     Dim Attach As String
60     Dim NotifText As String
61     Dim EmplIDRng As Range

```

1

```

1
60 Dim FoundEmplRng As Range
61 Dim EventRow As Long
62
63 Set EmplIDRng = Sheet2.Range("EmployeeID" )
64 UserName = Sheet14.Range("B10" ).Value 'Logged In User Name
65 UserEmail = Sheet14.Range("B11" ).Value 'Logged In user Email
66 With Sheet7
67     EventName = .Range("CB5" ).Value 'Event Name
68     EventType = .Range("CC5" ).Value 'Event Type
69     EventNotes = .Range("CG5" ).Value 'Event Notes
70     EventCreatedOn = .Range("CF5" ).Value 'Created On Date & Time
71     StartOn = .Range("CI5" ).Value 'Event Start On Date & Time
72     EndOn = .Range("CJ5" ).Value 'Event End On Date & Time
73     LeaveType = .Range("CU5" ).Value 'Leave Type
74     EmailTo = .Range("CP5" ).Value 'Email To
75     EmailBCC = .Range("CQ5" ).Value 'Email BCC
76     Subj = .Range("CS5" ).Value 'Email Subject
77     Mess = .Range("CT5" ).Value 'Email Message
78     Attach = .Range("CR5" ).Value 'Attachments
79     NotifText = .Range("CX5" ).Value 'Notification text (pulled from Admin Notifications
      Table)
80     EventRow = .Range("CW5" ).Value 'Event Row
81
82     EmplID = .Range("CE5" ).Value 'Employee ID
83     Set FoundEmplRng = EmplIDRng.Find(EmplID, , xlValues, xlWhole)
84     If Not FoundEmplRng Is Nothing Then 'Locate Employee via Employee ID
85         EmployeeName = Sheet2.Range("C" & FoundEmplRng.Row).Value 'Employee First Name
          (Change column for Full name or Last Name)
86         EmployeeEmail = Sheet2.Range("M" & FoundEmplRng.Row).Value 'Employee Email
87     End If
88
89     'Send Email If Applicable
90     If .Range("CN5" ).Value = "Yes" Then
91         EmailTo = Replace(Replace(EmailTo, "[User Email]" , UserEmail), "[Employee Email]" ,
          EmployeeEmail) 'Replace User & Employee Email as needed for Email To
92         EmailBCC = Replace(Replace(EmailBCC, "[User Email]" , UserEmail), "[Employee
          Email]" , EmployeeEmail) 'Replace User & Employee Email as needed for Email To
93         Subj = Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Subj,
          "[Event Name]" , EventName), "[Event Type]" , EventType), "[Event Notes]" ,
          EventNotes), "[Event Created On Date]" , EventCreatedOn), "[Start/Next Date &
          Time]" , StartOn), _
94         "[End Date & Time]" , EndOn), "[Employee Name]" , EmployeeName), "[User Name]" ,
          UserName), "[Leave Type]" , LeaveType)
95         Mess = Replace(Replace(Replace(Replace(Replace(Replace(Replace(Mess,
          "[Event Name]" , EventName), "[Event Type]" , EventType), "[Event Notes]" ,
          EventNotes), "[Event Created On Date]" , EventCreatedOn), "[Start/Next Date &
          Time]" , StartOn), _
96         "[End Date & Time]" , EndOn), "[Employee Name]" , EmployeeName), "[User Name]" ,
          UserName), "[Leave Type]" , LeaveType)
97
98     Set OutApp = CreateObject("Outlook.Application" ) 'Set the Application to Outlook
99     Set OutMail = OutApp.CreateItem(0) 'Crete a New Email
100    With OutMail
101        .To = EmailTo 'Send To Email
102        .BCC = EmailBCC
103        .Subject = Subj 'Adds in The Email Subject
104        .Body = Mess 'Adds in the Email Messages
105        If Attach <> "" Then .Attachments.Add Attach 'Attached the file (xlsx, pdf or
          picture file)
106        .Display 'Displays the email | Use .Send to send email without displaying
          first

```

1 2 3 4

```
1 2 3 4
107     End With
108     On Error GoTo 0
109     Set OutMail = Nothing 'Clear Outlook Email
110     End If
111     'Create Notification If Applicable
112     If .Range("CM5" ).Value = "Yes" Then
113         NotifText = Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(
114             NotifText, "[Event Name]" , EventName), "[Event Type]" , EventType), "[Event
115             Notes]" , EventNotes), "[Event Created On Date]" , EventCreatedOn), "[Start/Next
116             Date & Time]" , StartOn), _
117             "[End Date & Time]" , EndOn), "[Employee Name]" , EmployeeName), "[User Name]" ,
118             UserName), "[Leave Type]" , LeaveType)
119         .Range("CY5" ).Value = NotifText
120         .Range(5 & ":" & 5).WrapText = False
121         Sheet12.Shapes("NotificationGrp" ).Copy
122         ActiveSheet.Range("A1" ).Select
123         ActiveSheet.Paste
124         With ActiveSheet.Shapes("NotificationGrp" )
125             .Height = 172
126             .Width = 197
127             .Left = ActiveCell.Left
128             .Top = ActiveCell.Top
129             .Visible = msoCTrue
130         End With
131         ActiveSheet.Shapes("YellowNote" ).TextFrame.Characters.Text = NotifText
132         ActiveSheet.Range("A1" ).Select
133     End If
134     'Set Reminded On Date & time
135     .Range("R" & EventRow).Value = Now 'Set Reminded On Date & Time To Current Date &
136     Time
137 End With
138 Notifications_Check 'Recheck List
139 End Sub
140
141 Sub testting()
142     With ActiveSheet.Shapes("NotificationGrp" )
143         MsgBox "Height:" & .Height & vbCrLf & "Width:" & .Width
144     End With
145 End Sub
146
```

—, 119, 120

## A

ActiveCell, 120  
 ActiveSheet, 118, 120  
 Add, 118, 119  
 AdvancedFilter, 118  
 Apply, 118  
 Attach, 118, 119  
 Attachments, 119

## B

BCC, 119  
 Body, 119

## C

Characters, 120  
 Clear, 118  
 ClearContents, 118  
 Copy, 120  
 CopytoRange, 118  
 CreateItem, 119  
 CreateObject, 119  
 CriteriaRange, 118

## D

DataOption, 118  
 Delete, 118  
 Display, 119

## E

EmailBCC, 118, 119  
 EmailTo, 118, 119  
 EmplID, 118, 119  
 EmplIDRng, 118, 119  
 EmployeeEmail, 118, 119  
 EmployeeName, 118-120  
 EndOn, 118-120  
 EventCreatedOn, 118-120  
 EventName, 118-120  
 EventNotes, 118-120  
 EventRow, 118-120  
 EventType, 118-120  
 Explicit, 118

## F

Find, 119  
 FoundEmplRng, 119

## H

Height, 120

## K

Key, 118

## L

LastFiltRow, 118  
 LastRow, 118  
 LeaveType, 118-120  
 Left, 120

## M

Mess, 118, 119  
 MsgBox, 120  
 msoCTrue, 120

## N

NoNotif, 118

Notification\_Close, 118  
 Notifications\_Check, 118, 120  
 Notifications\_Create, 118  
 NotifText, 118-120  
 Now, 118, 120

## O

Order, 118  
 OutApp, 118, 119  
 OutMail, 118-120

## P

Paste, 120

## R

Range, 118-120  
 Replace, 119, 120  
 Row, 118, 119

## S

SetRange, 118  
 Shapes, 118, 120  
 Sheet12, 120  
 Sheet14, 119  
 Sheet2, 119  
 Sheet7, 118, 119  
 Sort, 118  
 SortFields, 118  
 SortOn, 118  
 StartOn, 118-120  
 Subj, 118, 119  
 Subject, 119

## T

testting, 120  
 Text, 120  
 TextFrame, 120  
 Top, 120

## U

Unique, 118  
 UserEmail, 118, 119  
 UserName, 118-120

## V

Value, 118-120  
 vbCrLf, 120  
 Visible, 120

## W

Width, 120  
 WrapText, 120

## X

xlAscending, 118  
 xlFilterCopy, 118  
 xlSortNormal, 118  
 xlSortOnValues, 118  
 xlUp, 118  
 xlValues, 119  
 xlWhole, 119

## Option Explicit

```

1  Option Explicit
2
3  Sub Payroll_Load_Review()
4      Dim PayrollRow As Long
5      Dim LastPayEmpRow As Long
6      Dim LastFiltRow
7      With Sheet16
8          If .Range("B15").Value = Empty Then Exit Sub ' Exit On No Payroll Row Found
9
10         .Range("B1").Value = True 'Set Payroll Load to True
11         .Range("B2").Value = .Range("B15").Value 'Set Payroll ID
12         StopCalc
13         PayrollRow = .Range("B3").Value 'Payroll Row
14         .Range("AJ8,AE14:AH44,AK9:AK44,AL14:AL25,AP14:AQ25,AL29:AM40,AP29:AQ40").ClearContents
15         'clear fields
16         .Range("AA9:AD310,AO14:A040,AS14:AS25,AW29:AW40").ClearContents 'clear fields
17         .Range("AM3").Value = Sheet18.Range("C" & PayrollRow).Value 'Payroll Date
18         .Range("J6").Value = Sheet18.Range("D" & PayrollRow).Value 'From Date
19         .Range("J8").Value = Sheet18.Range("E" & PayrollRow).Value 'To Date
20         .Range("AN3").Value = Sheet18.Range("F" & PayrollRow).Value 'Payroll Status
21         'Add Employees
22         Sheet20.Range("L3:Q9999").ClearContents 'Clear Prev Criteria & filters
23         Sheet20.Range("L3").Value = .Range("B2").Value 'Payroll ID
24         LastPayEmpRow = Sheet20.Range("A9999").End(xlUp).Row 'Last Employee Row
25         Sheet20.Range("A2:E" & LastPayEmpRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
26         Sheet20.Range("L2:L3"), CopytoRange:=Sheet20.Range("M2:Q2"), Unique:=True
27         LastFiltRow = Sheet20.Range("M9999").End(xlUp).Row
28         If LastFiltRow < 3 Then GoTo NoData
29         .Range("AA9:AB" & LastFiltRow + 6).Value = Sheet20.Range("M3:N" & LastFiltRow).Value
30         'Add Status & Employee ID
31         .Range("AD9:AD" & LastFiltRow + 6).Value = Sheet20.Range("Q3:Q" & LastFiltRow).Value
32         'Add Table Rows
33         .Range("AC7").Copy
34         .Range("AC9:AC" & LastFiltRow + 6).PasteSpecial xlPasteFormulas
35         .Calculate
36         .Range("AC9:AC" & LastFiltRow + 6).Copy
37         .Range("AC9").PasteSpecial xlPasteValues
38         Application.CutCopyMode = xlCopy
39         NoData:
40         .Range("B1").Value = False 'Set Payroll Load to False
41         ResetCalc
42         .Range("AC9").Select
43     End With
44 End Sub

```

```

42  Sub Payroll_Load_Summary()
43      Dim PayrollRow As Long
44      Dim LastPayEmpRow As Long
45      Dim LastFiltRow
46      With Sheet16
47          If .Range("B15").Value = Empty Then Exit Sub ' Exit On No Payroll Row Found
48
49         .Range("B1").Value = True 'Set Payroll Load to True
50         .Range("B2").Value = .Range("B16").Value 'Set Payroll ID
51         StopCalc
52         PayrollRow = .Range("B3").Value 'Payroll Row
53         .Range("BC9:BC999,BD10:BI999").ClearContents 'Clear fields
54         .Range("BD6").Value = Sheet18.Range("C" & PayrollRow).Value 'Payroll Date
55         .Range("BG6").Value = Sheet18.Range("D" & PayrollRow).Value 'From Date
56         .Range("BI6").Value = Sheet18.Range("E" & PayrollRow).Value 'To Date
57         .Range("BL3").Value = Sheet18.Range("F" & PayrollRow).Value 'Payroll Status

```



```
58      1 2 'Add Employees
59      Sheet20.Range("L3:Q9999" ).ClearContents 'Clear Prev Criteria & filters
60      Sheet20.Range("L3" ).Value = .Range("B2" ).Value 'Payroll ID
61      LastPayEmpRow = Sheet20.Range("A9999" ).End(xlUp).Row 'Last Employee Row
62      Sheet20.Range("A2:E" & LastPayEmpRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
63      Sheet20.Range("L2:L3" ), CopytoRange:=Sheet20.Range("N2" ), Unique:=True
64      LastFiltRow = Sheet20.Range("N9999" ).End(xlUp).Row
65      If LastFiltRow < 3 Then GoTo NoData
66      .Range("BC9:BC" & LastFiltRow + 6).Value = Sheet20.Range("N3:N" & LastFiltRow).Value
67      'Add Employee ID
68      .Range("BD9:BI" & LastFiltRow + 6).Formula = .Range("BD9:BI9" ).Formula 'Add Formulas
69
70      NoData:
71      .Range("B1" ).Value = False 'Set Payroll Load to False
72      ResetCalc
73      .Range("BD5" ).Select
74      End With
75      End Sub
```

## A

AdvancedFilter, [122](#), [123](#)  
Application, [122](#)

## C

Calculate, [122](#)  
ClearContents, [122](#), [123](#)  
Copy, [122](#)  
CopytoRange, [122](#), [123](#)  
CriteriaRange, [122](#), [123](#)  
CutCopyMode, [122](#)

## E

Empty, [122](#)  
Explicit, [122](#)

## F

Formula, [123](#)

## L

LastFilterRow, [122](#), [123](#)  
LastPayEmpRow, [122](#), [123](#)

## N

NoData, [122](#), [123](#)

## P

PasteSpecial, [122](#)  
Payroll\_Load\_Review, [122](#)  
Payroll\_Load\_Summary, [122](#)  
PayrollRow, [122](#)

## R

Range, [122](#), [123](#)  
ResetCalc, [122](#), [123](#)  
Row, [122](#), [123](#)

## S

Sheet16, [122](#)  
Sheet18, [122](#)  
Sheet20, [122](#), [123](#)  
StopCalc, [122](#)

## U

Unique, [122](#), [123](#)

## V

Value, [122](#), [123](#)

## X

xlCopy, [122](#)  
xlFilterCopy, [122](#), [123](#)  
xlPasteFormulas, [122](#)  
xlPasteValues, [122](#)  
xlUp, [122](#), [123](#)

```

1 Option Explicit
2
3 Sub Payroll_LoadEmpReview()
4 Dim ReviewRow As Long
5 Dim EmpRow As Long
6 Dim LastFiltRow As Long
7 Dim EmpID As String
8 Dim PayrollID As String
9 Dim LastTimeclockRow As Long
10 Dim LastPayItemRow As Long
11 Dim PayItemRow As Long
12 Dim PayrollRow As Long
13 Dim EmpPayRow As Long
14 Dim ItemType As String
15 StopCalc
16 With Sheet16
17 EmpRow = .Range("B10").Value 'Employee Table Row
18 EmpID = .Range("B11").Value 'Employee ID
19 ReviewRow = .Range("B12").Value 'Selected Payroll Review Employee Row
20 .Range("AJ8,AE14:AH44,AK9:AK44,AL14:AL25,AP14:AQ25,AL29:AM40,AP29:AQ40").ClearContents 'clear fields
21 .Range("A014:A040,AS14:AS25,AW29:AW40").ClearContents 'clear fields
22 .Range("AJ8").Value = Sheet2.Range("O" & EmpRow).Value 'Pay Type (Salary/Hourly)
23 If .Range("AJ8").Value = "Salary" Then 'Salary Employee
24 .Range("AH10").Value = Sheet2.Range("P" & EmpRow).Value 'Salary Frequency (if any)
25 Else: 'Hourly Employee
26 .Range("AH10").Value = Sheet2.Range("R" & EmpRow).Value 'Overtime Rate (if any)
27 End If
28 .Range("AH9").Value = Sheet2.Range("Q" & EmpRow).Value 'Salary/Hourly Amount
29
30 'Add in Time Clock Data
31 Sheet9.Range("AA3:AI3,AA5:AAI99999").ClearContents 'Clear any Previous Criteria & Results In Timeclock sheet
32 Sheet9.Range("AB3").Value = EmpID 'Employee ID
33 Sheet9.Range("Z3").Value = ">=" & .Range("PayrollStart").Value 'From Payroll Date
34 Sheet9.Range("AD3").Value = "<=" & .Range("PayrollEnd").Value 'To Payroll Date
35 LastTimeclockRow = Sheet9.Range("A99999").End(xlUp).Row
36 Sheet9.Range("A4:I" & LastTimeclockRow).AdvancedFilter xlFilterCopy, CriteriaRange:=Sheet9.Range("Z2:AD3"), CopytoRange:=Sheet9.Range("AL4:AP4"), Unique:=True
37 LastFiltRow = Sheet9.Range("AL99999").End(xlUp).Row
38 If LastFiltRow < 5 Then GoTo NoTimeclock
39 'Sort TimeClock Entries Newest To Oldest
40 Sheet9.Sort.SortFields.Clear
41 Sheet9.Sort.SortFields.Add Key:=.Range("AL5"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
42 With Sheet9.Sort
43 .SetRange Range("AL5:AP" & LastFiltRow)
44 .Apply
45 End With
46 If LastFiltRow > 35 Then LastFiltRow = 35 'Set Maximum of 31 Rows
47 .Range("AE14:AH" & LastFiltRow + 9).Value = Sheet9.Range("AL5:A0" & LastFiltRow).Value 'Copy Clock In/Out & Breaks
48 .Range("AK14:AK" & LastFiltRow + 9).Value = Sheet9.Range("AP5:AP" & LastFiltRow).Value 'Copy Table Row #'s
49 NoTimeclock:
50 'Add in Taxes, Other Deductions, Other Additions & Leave Based on Saved/Pending Status
51 If .Range("AA" & ReviewRow).Value = "6" Then 'Pending Status (pull from Employee Info)
52

```

1 2 3

```

1 2 3
53 Sheet17.Range("M2:W99999").ClearContents 'Clear any Previous Criteria & Results
54 Sheet17.Range("M2").Value = EmpID 'Employee ID
55 LastPayItemRow = Sheet17.Range("A99999").End(xlUp).Row
56 Sheet17.Range("A2:I" & LastPayItemRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
Sheet17.Range("M1:M2"), CopytoRange:=Sheet17.Range("O1:W1"), Unique:=True
57 LastFiltRow = Sheet17.Range("O99999").End(xlUp).Row
58 If LastFiltRow < 2 Then GoTo NoPayItem
59 For PayItemRow = 2 To LastFiltRow
60     ItemType = Sheet17.Range("P" & PayItemRow).Value 'Pay Item Type
61     EmpPayRow = Sheet17.Range("V" & PayItemRow).Value 'Employee Manager Row
62
63     If ItemType = "Tax Deduction" Then 'Tax Deduction Items (Row -79)
64         .Range("AL" & EmpPayRow - 79).Value = Sheet17.Range("Q" & PayItemRow).Value
        'Item Name
65     End If
66     If ItemType = "Pay Deduction" Then
67         .Range("AP" & EmpPayRow - 94).Value = Sheet17.Range("Q" & PayItemRow).Value
        'Item Name
68         .Range("AQ" & EmpPayRow - 94).Value = Sheet17.Range("T" & PayItemRow).Value
        'Amount
69     End If
70     If ItemType = "Pay Addition" Then
71         .Range("AL" & EmpPayRow - 99).Value = Sheet17.Range("Q" & PayItemRow).Value
        'Item Name
72         .Range("AM" & EmpPayRow - 99).Value = Sheet17.Range("T" & PayItemRow).Value
        'Amount
73     End If
74 Next PayItemRow
75 NoPayItem:
76 'Add In Employee Leave
77 Dim LastLeaveRow As Long
78 Sheet11.Range("AA3,AB3,AA5:ADI99999").ClearContents 'Clear any Previous Results
79 Sheet11.Range("AA3").Value = EmpID 'Employee ID
80 Sheet11.Range("AB3").Value = "Paid" 'Only Filter Paid Leave Items
81 LastLeaveRow = Sheet11.Range("A99999").End(xlUp).Row
82 Sheet11.Range("A4:H" & LastLeaveRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
Sheet11.Range("AA2:AB3"), CopytoRange:=Sheet11.Range("AA4:AE4"), Unique:=True
83 LastFiltRow = Sheet11.Range("AA99999").End(xlUp).Row
84 If LastFiltRow < 5 Then GoTo NoLeave
85
86 If LastFiltRow > 16 Then LastFiltRow = 16 'Set Maximum of 12 Rows
87 .Range("AP29:AP" & LastFiltRow + 24).Value = Sheet11.Range("AA5:AA" & LastFiltRow)
    .Value 'Leave Name
88 NoLeave:
89
90 } Else: 'Previously Saved Employee Details (Pull from Payroll Item List)
91 'Run Advanced Filter on Payroll Items
92 PayrollID = .Range("B2").Value 'Payroll ID
93 Sheet19.Range("M3:V99999").ClearContents 'Clear Previous Criteria & Results
94 Sheet19.Range("M3").Value = PayrollID 'Set Payroll ID
95 Sheet19.Range("N3").Value = EmpID 'Set Employee ID
96 LastPayItemRow = Sheet19.Range("A99999").End(xlUp).Row
97 If LastPayItemRow < 3 Then GoTo NoPayData
98 Sheet19.Range("A2:J" & LastPayItemRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
Sheet19.Range("M2:N3"), CopytoRange:=Sheet19.Range("P2:V2"), Unique:=True
99 LastFiltRow = Sheet19.Range("P999").End(xlUp).Row
100 If LastFiltRow < 3 Then GoTo NoPayData
101 For PayItemRow = 3 To LastFiltRow
102     ItemType = Sheet19.Range("P" & PayItemRow).Value 'Set Item Type
103     PayrollRow = Sheet19.Range("U" & PayItemRow).Value 'Set Payroll row
104     If ItemType = "Regular Pay" Then .Range("AK9").Value = Sheet19.Range("V" &

```

1 2 3 4

```

1 2 3 4
105 PayItemRow).Value 'Table Row For Update
    If ItemType = "OT Pay" Then .Range("AK10" ).Value = Sheet19.Range("V" &
PayItemRow).Value 'Table Row For Update
106 If ItemType = "Salary Pay" Then .Range("AK10" ).Value = Sheet19.Range("V" &
PayItemRow).Value 'Table Row For Update
107 If ItemType = "Tax Item" Then
108     .Range("AL" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
    'Tax Item Name
109     .Range("AO" & PayrollRow).Value = Sheet19.Range("V" & PayItemRow).Value
    'Pay Item Table Row
110 End If
111 If ItemType = "Other Deduction" Then
112     .Range("AP" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
    'Deduction Item Name
113     .Range("AQ" & PayrollRow).Value = Sheet19.Range("R" & PayItemRow).Value
    'Deduction Amount
114     .Range("AS" & PayrollRow).Value = Sheet19.Range("V" & PayItemRow).Value
    'Pay Item Table Row
115 End If
116 If ItemType = "Other Addition" Then
117     .Range("AL" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
    'Addition Item Name
118     .Range("AM" & PayrollRow).Value = Sheet19.Range("R" & PayItemRow).Value
    'Addition Amount
119     .Range("AO" & PayrollRow).Value = Sheet19.Range("V" & PayItemRow).Value
    'Pay Item Table Row
120 End If
121 If ItemType = "Paid Leave" Then
122     .Range("AP" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
    'Leave Item Name
123     .Range("AQ" & PayrollRow).Value = Sheet19.Range("S" & PayItemRow).Value
    'Leave Hours
124     .Range("AW" & PayrollRow).Value = Sheet19.Range("V" & PayItemRow).Value
    'Pay Item Table Row
125 End If
126 Next PayItemRow
127 NoPayData:
128 End If
129
130 End With
131 ResetCalc
132 End Sub
133
134 Sub Payroll_Review_EditTimeClockEntry()
135 Dim SelRow As Long
136 Dim DateRow As Long
137 Dim TCRow As Long
138 Dim ClockInVal As Double
139 Dim ClockOutVal As Double
140 Dim BreakOutVal As Double
141 Dim BreakInVal As Double
142
143 With Sheet16
144     SelRow = .Range("B14" ).Value 'Selected Row
145     TCRow = .Range("AK" & SelRow).Value 'Time Clock Row
146     Sheet8.Range("A53" ).Value = .Range("B11" ).Value 'Add Employee ID
147     Sheet8.Range("A57" ).Value = Int(.Range("AE" & SelRow)) 'Add Selected Date
148
149     ClockInVal = Sheet9.Range("D" & TCRow).Value 'Clock In
150     BreakOutVal = Sheet9.Range("E" & TCRow).Value 'Break out
151     BreakInVal = Sheet9.Range("F" & TCRow).Value 'Break In

```

1 2

```
1 2
152 ClockOutVal = Sheet9.Range("G" & TCRow).Value 'Clock In
153 With EditTimeClock
154     If ClockInVal = 0 Then .ClockIn.Value = "" Else: .ClockIn.Value = ClockInVal
        'Clock In
155     If BreakOutVal = 0 Then .BreakOut.Value = "" Else: .BreakOut.Value = BreakOutVal
        'Clock Out Break
156     If BreakInVal = 0 Then .BreakIn.Value = "" Else: .BreakIn.Value = BreakInVal
        'Clock In Break
157     If ClockOutVal = 0 Then .ClockOut.Value = "" Else: .ClockOut.Value = ClockOutVal
        'Clock Out
158     .Notes.Value = Sheet9.Range("H" & TCRow).Value 'Notes
159 End With
160 EditTimeClock_Calculate
161 EditTimeClock.Show 'Display Edit Time Clock form
162
163 End With
164
165 End Sub
166
167 Sub Payroll_ShowEditEmployeeIcon()
168     With Sheet16
169         Dim SelRow As Long
170         If .Range("B12").Value = Empty Then Exit Sub
171         SelRow = .Range("B12").Value
172         With .Shapes("EditEmployeeBtn")
173             .Left = Sheet16.Range("AD" & SelRow).Left
174             .Top = Sheet16.Range("AD" & SelRow).Top
175             .Visible = msoCTrue
176         End With
177     End With
178 End Sub
179 Sub Payroll_ShowEditTimeClockIcon()
180     With Sheet16
181         Dim SelRow As Long
182         If .Range("B14").Value = Empty Then Exit Sub
183         SelRow = .Range("B14").Value
184         With .Shapes("EditTimeClockBtn")
185             .Left = Sheet16.Range("AE" & SelRow).Left
186             .Top = Sheet16.Range("AE" & SelRow).Top
187             .Visible = msoCTrue
188         End With
189     End With
190 End Sub
191
192 Sub Payroll_ViewEmployeeInfo()
193     Dim EmpID As String
194     With Sheet16
195         If .Range("AB" & .Range("B12").Value).Value = Empty Then Exit Sub
196         EmpID = .Range("AB" & .Range("B12").Value).Value 'Employee ID
197         Sheet1.Activate
198         Sheet1.Range("J2").Value = EmpID
199     End With
200 End Sub
```

**A**

Activate, 128  
Add, 125  
AdvancedFilter, 125, 126  
Apply, 125

**B**

BreakIn, 128  
BreakInVal, 127, 128  
BreakOut, 128  
BreakOutVal, 127, 128

**C**

Clear, 125  
ClearContents, 125, 126  
ClockIn, 128  
ClockInVal, 127, 128  
ClockOut, 128  
ClockOutVal, 127, 128  
CopytoRange, 125, 126  
CriteriaRange, 125, 126

**D**

DataOption, 125  
DateRow, 127

**E**

EditTimeClock, 128  
EditTimeClock\_Calculate, 128  
EmpID, 125, 126, 128  
EmpPayRow, 125, 126  
EmpRow, 125  
Empty, 128  
Explicit, 125

**I**

Int, 127  
ItemType, 125-127

**K**

Key, 125

**L**

LastFiltRow, 125, 126  
LastLeaveRow, 126  
LastPayItemRow, 125, 126  
LastTimeclockRow, 125  
Left, 128

**M**

msoCTrue, 128

**N**

NoLeave, 126  
NoPayData, 126, 127  
NoPayItem, 126  
Notes, 128  
NoTimeclock, 125

**O**

Order, 125

**P**

PayItemRow, 125-127  
Payroll\_LoadEmpReview, 125  
Payroll\_Review\_EditTimeClockEntry, 127  
Payroll\_ShowEditEmployeeIcon, 128  
Payroll\_ShowEditTimeClockIcon, 128  
Payroll\_ViewEmployeeInfo, 128

PayrollID, 125, 126  
PayrollRow, 125-127

**R**

Range, 125-128  
ResetCalc, 127  
ReviewRow, 125  
Row, 125, 126

**S**

SelRow, 127, 128  
SetRange, 125  
Shapes, 128  
Sheet1, 128  
Sheet11, 126  
Sheet16, 125, 127, 128  
Sheet17, 126  
Sheet19, 126, 127  
Sheet2, 125  
Sheet8, 127  
Sheet9, 125, 127, 128  
Show, 128  
Sort, 125  
SortFields, 125  
SortOn, 125  
StopCalc, 125

**T**

TCRow, 127, 128  
Top, 128

**U**

Unique, 125, 126

**V**

Value, 125-128  
Visible, 128

**X**

xlAscending, 125  
xlFilterCopy, 125, 126  
xlSortNormal, 125  
xlSortOnValues, 125  
xlUp, 125, 126

```
1 Option Explicit
```

```
4 Sub Payroll_PrevEmployee()
```

```
5 With Sheet16
```

```
6 If .Range("B12").Value = 9 Then
```

```
7     MsgBox "You are at the first employee"
```

```
8     Exit Sub
```

```
9 End If
```

```
10 .Range("AC" & .Range("B12").Value - 1).Select
```

```
11 .Range("AL3").Select 'Select another cell on sheet
```

```
12 End With
```

```
13 End Sub
```

```
15 Sub Payroll_SaveEmployeePay()
```

```
16 Dim PayrollID As String
```

```
17 Dim EmpID As String
```

```
18 Dim PayType As String
```

```
19 Dim ReviewRow As Long
```

```
20 Dim PayrollListRow As Long
```

```
21 Dim PayrollRow As Long
```

```
22 Dim PayItemRow As Long
```

```
23 Dim PayEmpListRow As Long
```

```
24 Dim PayrollEmpRow As Long
```

```
25 Dim LastPayrollEmpRow As Long
```

```
26 With Sheet16
```

```
27     StopCalc
```

```
29     'Check To Make sure Payroll has been saved, if not, save it
```

```
30     If .Range("B2").Value = Empty Then 'New Payroll
```

```
31         .Range("B2").Value = .Range("B5").Value 'Set Next Payroll ID
```

```
32         PayrollID = .Range("B2").Value
```

```
33         PayrollListRow = Sheet18.Range("A9999").End(xlUp).Row + 1 'First Available  
Payroll Row
```

```
34         Sheet18.Range("A" & PayrollListRow).Value = PayrollID 'Next Payroll ID
```

```
35         Sheet18.Range("B" & PayrollListRow).Value = Application.WorksheetFunction.Text(.  
Range("PayrollStart").Value, "mm/dd/yyyy") & "_" & Application.WorksheetFunction.  
Text(.Range("PayrollEnd").Value, "mm/dd/yyyy") 'Set Payroll Range
```

```
36         Sheet18.Range("C" & PayrollListRow).Value = .Range("AM3").Value 'Payroll Date
```

```
37         Sheet18.Range("D" & PayrollListRow).Value = .Range("PayrollStart").Value 'Payroll  
Start Date
```

```
38         Sheet18.Range("E" & PayrollListRow).Value = .Range("PayrollEnd").Value 'Payroll  
End Date
```

```
39         Sheet18.Range("F" & PayrollListRow).Value = "Pending Review" 'Status
```

```
40         .Range("AN3").Value = "Pending Review"
```

```
41         .Range("AL3").Value = Sheet18.Range("B" & PayrollListRow).Value 'Set Payroll Date  
Range
```

```
42         'Save Employee To Pay List
```

```
43         PayEmpListRow = Sheet20.Range("A9999").End(xlUp).Row + 1
```

```
44         LastPayrollEmpRow = .Range("AB999").End(xlUp).Row 'Last Employee To Be Paid Row
```

```
46     For PayrollEmpRow = 9 To LastPayrollEmpRow
```

```
47         Sheet20.Range("A" & PayEmpListRow).Value = PayrollID 'Payroll ID
```

```
48         Sheet20.Range("B" & PayEmpListRow).Value = .Range("AB" & PayrollEmpRow).Value  
'Employee ID
```

```
49         Sheet20.Range("C" & PayEmpListRow).Value = .Range("AA" & PayrollEmpRow).Value  
'Review Status
```

```
50         Sheet20.Range("D" & PayEmpListRow).Value = PayrollEmpRow 'Payroll Row
```

```
51         Sheet20.Range("E" & PayEmpListRow).Value = "=Row()" 'Add in Sheet Row Formula
```

```
52         .Range("AD" & PayrollEmpRow).Value = PayEmpListRow 'Set Employee List Row
```

```
53         PayEmpListRow = PayEmpListRow + 1 'Add one for Next Employee Payroll List row
```

```
1 2 3 4
```



```

54     1 2 3 4
55     Next PayrollEmpRow
56     Else: 'Payroll Previously Saved
57     PayrollListRow = .Range("B3" ).Value 'Payroll Row
58     End If
59     ReviewRow = .Range("B12" ).Value 'Current Employee Reivew Row
60     PayrollID = .Range("B2" ).Value
61     EmpID = .Range("B11" ).Value 'Employee ID
62     'Add In Hourly or Salary Income Items
63     PayType = .Range("AJ8" ).Value 'Set Pay Type
64     If PayType = "Hourly" Then 'Add in Regular & OT Amounts (If applicable)
65     If .Range("AK9" ).Value = Empty Then 'New Reg. Pay Item
66     PayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row + 1 'First Available Row
67     Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
68     Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
69     Sheet19.Range("C" & PayItemRow).Value = "Regular Pay" 'Set Pay Item Type
70     Sheet19.Range("D" & PayItemRow).Value = "Regular Pay" 'Set Pay Item Name
71     Sheet19.Range("G" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
72     PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
73     locate Payroll Date
74     Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
75     Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
76     .Range("AK9" ).Value = PayItemRow 'Set Pay Item Row
77     Else: 'Existing Regular Pay Item
78     PayItemRow = .Range("AK9" ).Value 'Existing Pay Item Row
79     End If
80     Sheet19.Range("E" & PayItemRow).Value = .Range("AJ9" ).Value 'Set Regular Hourly
81     Pay Amount
82     Sheet19.Range("F" & PayItemRow).Value = .Range("AF9" ).Value 'Set Regular Hours
83     If .Range("AJ10" ).Value <> 0 Then 'Add in OT Amount
84     If .Range("AK10" ).Value = Empty Then 'New Pay Item
85     PayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row + 1 'First Available Row
86     Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
87     Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
88     Sheet19.Range("C" & PayItemRow).Value = "OT Pay" 'Set Pay Item Type
89     Sheet19.Range("D" & PayItemRow).Value = "OT Pay" 'Set Pay Item Type
90     Sheet19.Range("G" & PayItemRow).Value =
91     "=IFERROR(INDEX(Payroll_Date,MATCH(A" & PayItemRow & ",Payroll_ID,0),1)," &
92     Chr(34) & Chr(34) & ")" 'Use formula to locate Payroll Date
93     Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
94     Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
95     .Range("AK10" ).Value = PayItemRow 'Set Pay Item Row
96     Else: 'Existing OT Pay Item
97     PayItemRow = .Range("AK10" ).Value 'Existing Pay Item Row
98     End If
99     Sheet19.Range("E" & PayItemRow).Value = .Range("AJ10" ).Value 'Set OT Pay
100    Amount
101    Sheet19.Range("F" & PayItemRow).Value = .Range("AF10" ).Value 'Set OT Hours
102    End If ' End Of OT Amount Add in
103    Else: 'Salary
104    If .Range("AK10" ).Value = Empty Then 'New Salary Pay Item
105    PayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row + 1 'First Available Row
106    Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
107    Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
108    Sheet19.Range("C" & PayItemRow).Value = "Salary Pay" 'Set Pay Item Type
109    Sheet19.Range("D" & PayItemRow).Value = "Salary Pay" 'Set Pay Item Type
110    Sheet19.Range("G" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
111    PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
112    locate Payroll Date
113    Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"

```

1 2 3 4

```

1 2 3 4
107 Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
108 .Range("AK10" ).Value = PayItemRow 'Set Pay Item Row
109 Else: 'Existing OT Pay Item
110 PayItemRow = .Range("AK10" ).Value 'Existing Pay Item Row
111 End If
112 Sheet19.Range("E" & PayItemRow).Value = .Range("AJ10" ).Value 'Set Salary Pay
113 Amount
114 End If 'End of Hourly & Salary
115 'Add In Tax Items
116 For PayrollRow = 14 To 25
117 If .Range("AL" & PayrollRow).Value = Empty Then Exit For
118 If .Range("AO" & PayrollRow).Value = Empty Then
119 PayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row + 1 'First Available Pay
120 Item Row
121 Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
122 Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
123 Sheet19.Range("C" & PayItemRow).Value = "Tax Item" 'Tax Item Type
124 Sheet19.Range("G" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
125 PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
126 locate Payroll Date
127 Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
128 Sheet19.Range("I" & PayItemRow).Value = PayrollRow
129 Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
130 .Range("AO" & PayrollRow).Value = PayItemRow
131 Else: 'Set Existing Pay Item Row
132 PayItemRow = .Range("AO" & PayrollRow).Value
133 End If
134 Sheet19.Range("D" & PayItemRow).Value = .Range("AL" & PayrollRow).Value 'Tax Item
135 Name
136 Sheet19.Range("E" & PayItemRow).Value = .Range("AM" & PayrollRow).Value 'Tax Item
137 Amount
138 Next PayrollRow
139 'Add In Other Payroll Deduction Items
140 For PayrollRow = 14 To 25
141 If .Range("AP" & PayrollRow).Value = Empty Then Exit For
142 If .Range("AS" & PayrollRow).Value = Empty Then
143 PayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row + 1 'First Available Pay
144 Item Row
145 Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
146 Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
147 Sheet19.Range("C" & PayItemRow).Value = "Other Deduction" 'Set Other Deduction
148 Item Type
149 Sheet19.Range("G" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
150 PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
151 locate Payroll Date
152 Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
153 Sheet19.Range("I" & PayItemRow).Value = PayrollRow
154 Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
155 .Range("AS" & PayrollRow).Value = PayItemRow
156 Else: 'Set Existing Pay Item Row
157 PayItemRow = .Range("AS" & PayrollRow).Value
158 End If
159 Sheet19.Range("D" & PayItemRow).Value = .Range("AP" & PayrollRow).Value 'Other
160 Deduction Name
161 Sheet19.Range("E" & PayItemRow).Value = .Range("AQ" & PayrollRow).Value 'Other
162 Deduction Amount
163 Next PayrollRow
164 'Add In Other Payroll Addition Items

```

```

156   1 2 For PayrollRow = 29 To 40
157     If .Range("AL" & PayrollRow).Value = Empty Then Exit For
158     If .Range("AO" & PayrollRow).Value = Empty Then
159       PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Pay
160       Item Row
161       Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
162       Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
163       Sheet19.Range("C" & PayItemRow).Value = "Other Addition" 'Set Other Addition
164       Item Type
165       Sheet19.Range("G" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
166         PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
167       locate Payroll Date
168       Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
169       Sheet19.Range("I" & PayItemRow).Value = PayrollRow
170       Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
171       .Range("AO" & PayrollRow).Value = PayItemRow
172     Else: 'Set Existing Pay Item Row
173       PayItemRow = .Range("AO" & PayrollRow).Value
174     End If
175     Sheet19.Range("D" & PayItemRow).Value = .Range("AL" & PayrollRow).Value 'Other
176     Deduction Name
177     Sheet19.Range("E" & PayItemRow).Value = .Range("AM" & PayrollRow).Value 'Other
178     Deduction Amount
179   Next PayrollRow
180
181   'Add In Employee Paid Leave Items
182   For PayrollRow = 29 To 40
183     If .Range("AP" & PayrollRow).Value = Empty Then Exit For
184     If .Range("AQ" & PayrollRow).Value <> Empty Then 'Only add items when they contain
185     hours
186     If .Range("AW" & PayrollRow).Value = Empty Then
187       PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Pay
188       Item Row
189       Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
190       Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
191       Sheet19.Range("C" & PayItemRow).Value = "Paid Leave" 'Set Other Addition
192       Item Type
193       Sheet19.Range("G" & PayItemRow).Value =
194         "=IFERROR(INDEX(Payroll_Date,MATCH(A" & PayItemRow & ",Payroll_ID,0),1)," &
195         Chr(34) & Chr(34) & ")" 'Use formula to locate Payroll Date
196       Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
197       Sheet19.Range("I" & PayItemRow).Value = PayrollRow
198       Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
199       .Range("AU" & PayrollRow).Value = PayItemRow
200     Else: 'Set Existing Pay Item Row
201       PayItemRow = .Range("AW" & PayrollRow).Value
202     End If
203     Sheet19.Range("D" & PayItemRow).Value = .Range("AP" & PayrollRow).Value 'Paid
204     Leave Name
205     Sheet19.Range("F" & PayItemRow).Value = .Range("AQ" & PayrollRow).Value 'Paid
206     Leave Hours
207   End If 'Check for Hours To Use in AQ
208   Next PayrollRow
209   .Range("AA" & ReviewRow).Value = "ü" 'Add Completed Checkbox
210   If .Range("AD" & ReviewRow).Value <> Empty Then Sheet20.Range("C" & .Range("AD" &
211     ReviewRow).Value).Value = "ü" 'Update Status in Employee Payroll List
212   ResetCalc
213
214   If .Range("B19").Value = 1 Then 'Update Status on All Employees Reviewed
215     .Range("AN3").Value = "Pending Confirmation"
216     Sheet18.Range("F" & PayrollListRow).Value = "Pending Confirmation"

```

1 2 3

```
203     1 2 3 End If
204     If .Range("AB" & ReviewRow + 1).Value <> Empty Then
205         .Range("AB" & ReviewRow + 1).Select
206     Else:
207         MsgBox "You have reached the last Employee"
208     End If
209     '.Range("AL3").Select 'Select another cell on sheet
210 End With
211 End Sub
```

**A**

Application, [130](#)

**C**

Chr, [131-133](#)

**E**

EmpID, [130-133](#)

Empty, [130-134](#)

Explicit, [130](#)

**L**

LastPayrollEmpRow, [130](#)

**M**

MsgBox, [130](#), [134](#)

**P**

PayEmpListRow, [130](#)

PayItemRow, [130-133](#)

Payroll\_PrevEmployee, [130](#)

Payroll\_SaveEmployeePay, [130](#)

PayrollEmpRow, [130](#), [131](#)

PayrollID, [130-133](#)

PayrollListRow, [130](#), [131](#), [133](#)

PayrollRow, [130](#), [132](#), [133](#)

PayType, [130](#), [131](#)

**R**

Range, [130-134](#)

ResetCalc, [133](#)

ReviewRow, [130](#), [131](#), [133](#), [134](#)

Row, [130-133](#)

**S**

Sheet16, [130](#)

Sheet18, [130](#), [133](#)

Sheet19, [131-133](#)

Sheet20, [130](#), [133](#)

StopCalc, [130](#)

**T**

Text, [130](#)

**V**

Value, [130-134](#)

**W**

WorksheetFunction, [130](#)

**X**

xlUp, [130-133](#)

```

1  Option Explicit
2
3  Sub Payroll_Details_To_Review()
4      Dim LastEmpRow As Long
5      With Sheet16
6          If .Range("M11" ).Value = Empty Then 'Check for No Employees
7              MsgBox "Please make sure you have at least one employee within the Employees To Be
              Paid List"
8              Exit Sub
9          End If
10         LastEmpRow = .Range("M999" ).End(xlUp).Row 'Last Employee To Pay Row
11         .Range("AA9:AD310" ).ClearContents
12         .Range("AB9:AC" & LastEmpRow - 2).Value = .Range("M11:N" & LastEmpRow).Value 'Copy
              Over Employees to be Paid
13         .Range("AA9:AA" & LastEmpRow - 2).Value = "6" 'Set Pending Status For Each Employee
14         .Range("B2" ).ClearContents 'Clear Previous Payroll ID
15         .Range("B12" ).Value = 9 'Set First Employee To Review
16         .Range("B11" ).Value = .Range("AB9" ).Value 'Set First Employee ID
17         .Range("AM3" ).Value = Date 'Set Current Date as Default Pay Date
18         .Range("AN3" ).Value = "Pending Review"
19         'Payroll_LoadEmpReview 'Run Load Employee Review Macro for First Employee
20
21         Payroll_SelectReviewSection
22         ResetCalc
23     End With
24 End Sub
25
26 Sub Payroll_SelectDetailsSection()
27     Dim LastEmpRow As Long
28     StopCalc
29     With Sheet16
30         .Range("C:Y" ).EntireColumn.Hidden = False
31         .Range("Z:BY" ).EntireColumn.Hidden = True
32         .Shapes("PayDetailGrp" ).Visible = msoTrue
33         .Shapes("PayDetailOptions" ).Visible = msoCTrue
34         .Shapes("PayReviewGrp" ).Visible = msoFalse
35         .Shapes("PaySummPendingGrp" ).Visible = msoFalse
36         .Shapes("PaySummConfirmedGrp" ).Visible = msoFalse
37         ActiveWindow.ScrollColumn = 3
38         ResetCalc
39     End With
40
41 End Sub
42 Sub Payroll_SelectReviewSection()
43     Dim PayrollRow As Long
44     With Sheet16
45         StopCalc
46         'On Change of Tab Update Payroll Date & Status if Payroll has been saved already
47         If .Range("B3" ).Value <> Empty Then
48             PayrollRow = .Range("B3" ).Value
49             .Range("AM3" ).Value = Sheet18.Range("C" & PayrollRow).Value 'Payroll Date
50             .Range("AN3" ).Value = Sheet18.Range("F" & PayrollRow).Value 'Payroll Status
51         End If
52         .Range("AZ:BY" ).EntireColumn.Hidden = True
53         .Range("C:Y" ).EntireColumn.Hidden = True
54         .Range("Z:AY" ).EntireColumn.Hidden = False
55         .Shapes("PayDetailGrp" ).Visible = msoFalse
56         .Shapes("PayDetailOptions" ).Visible = msoFalse
57         .Shapes("PayReviewGrp" ).Visible = msoTrue
58         .Shapes("PaySummPendingGrp" ).Visible = msoFalse
59         .Shapes("PaySummConfirmedGrp" ).Visible = msoFalse

```

```
1 2
60 ActiveWindow.ScrollColumn = 26
61 ResetCalc
62 End With
63 End Sub
64
65 Sub Payroll_SelectSummarySection()
66 Dim LastEmpRow As Long
67 With Sheet16
68 'Check for any remaining or pending employees
69 If .Range("B19").Value <> 1 Then 'Incomplete Reviews
70 MsgBox "All employees have not been reviewed yet. Please review each employee
before moving to the Summary & Confirmation section"
71 Exit Sub
72 End If
73 StopCalc
74 LastEmpRow = .Range("AB999").End(xlUp).Row 'Last Row of Employees
75 .Range("BC9:BC310").ClearContents 'Clear Any Existing Employees
76 .Range("BC9:BC" & LastEmpRow).Value = .Range("AB9:AB" & LastEmpRow).Value 'Copy Over
Employee ID's
77 .Range("BD6").Value = Sheet18.Range("C" & .Range("B3").Value).Value 'Payroll Date
78 .Range("BL3").Value = Sheet18.Range("F" & .Range("B3").Value).Value 'Payroll Status
79
80 If .Range("BL3").Value = "Pending Confirmation" Then
81 .Shapes("PaySummConfirmedGrp").Visible = msoFalse
82 .Shapes("PaySummPendingGrp").Visible = msoCTrue
83 Else:
84 .Shapes("PaySummConfirmedGrp").Visible = msoCTrue
85 .Shapes("PaySummPendingGrp").Visible = msoFalse
86 End If
87 .Range("BG6").Value = .Range("J6").Value 'From Date
88 .Range("BI6").Value = .Range("J8").Value 'To Date
89 .Range("C:AY").EntireColumn.Hidden = True
90 .Range("AZ:BY").EntireColumn.Hidden = False
91 .Shapes("PayDetailGrp").Visible = msoFalse
92 .Shapes("PayDetailOptions").Visible = msoFalse
93 .Shapes("PayReviewGrp").Visible = msoFalse
94 ActiveWindow.ScrollColumn = 52
95 Payroll_LoadPayrollSummary
96 ResetCalc
97 End With
98 End Sub
```

## A

ActiveWindow, [136](#), [137](#)

## C

ClearContents, [136](#), [137](#)

## E

Empty, [136](#)  
EntireColumn, [136](#), [137](#)  
Explicit, [136](#)

## H

Hidden, [136](#), [137](#)

## L

LastEmpRow, [136](#), [137](#)

## M

MsgBox, [136](#), [137](#)  
msoCTrue, [136](#), [137](#)  
msoFalse, [136](#), [137](#)  
msoTrue, [136](#)

## P

Payroll\_Details\_To\_Review, [136](#)  
Payroll\_LoadPayrollSummary, [137](#)  
Payroll\_SelectDetailsSection, [136](#)  
Payroll\_SelectReviewSection, [136](#)  
Payroll\_SelectSummarySection, [137](#)  
PayrollRow, [136](#)

## R

Range, [136](#), [137](#)  
ResetCalc, [136](#), [137](#)  
Row, [136](#), [137](#)

## S

ScrollColumn, [136](#), [137](#)  
Shapes, [136](#), [137](#)  
Sheet16, [136](#), [137](#)  
Sheet18, [136](#), [137](#)  
StopCalc, [136](#), [137](#)

## V

Value, [136](#), [137](#)  
Visible, [136](#), [137](#)

## X

xlUp, [136](#), [137](#)



```
1 Option Explicit
```

```
4 Sub Payroll_AddSelectedEmpl()
```

```
5 Dim PaidRow As Long
```

```
6 Dim SelRow As Long
```

```
7 Dim LastPaidRow As Long
```

```
8 Dim LastSelRow As Long
```

```
9 Dim FoundEmpIDRng As Range
```

```
10 With Sheet16
```

```
11 StopCalc
```

```
12 LastSelRow = .Range("D999").End(xlUp).Row
```

```
13 LastPaidRow = .Range("L999").End(xlUp).Row
```

```
14 If LastPaidRow < 10 Then LastPaidRow = 10
```

```
15 For SelRow = 11 To LastSelRow
```

```
16 If .Range("D" & SelRow).Value = "p" Then
```

```
17 'Check for Duplicates
```

```
18 Set FoundEmpIDRng = .Range("M11:M" & LastPaidRow).Find(.Range("E" & SelRow).Value, , xlValues, xlWhole)
```

```
19 If FoundEmpIDRng Is Nothing Then 'Not Found
```

```
20 LastPaidRow = LastPaidRow + 1
```

```
21 .Range("L" & LastPaidRow & ":N" & LastPaidRow).Value = .Range("D" & SelRow & ":F" & SelRow).Value 'Move Employee To First Available Row
```

```
22 End If
```

```
23 .Range("D" & SelRow & ":H" & SelRow).ClearContents 'Clear Moved Over values
```

```
24 End If
```

```
25 Next SelRow
```

```
26 .Sort.SortFields.Clear
```

```
27 .Sort.SortFields.Add Key:=.Range("F11"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
```

```
28 With Sheet16.Sort
```

```
29 .SetRange Range("D11:F" & LastSelRow)
```

```
30 .Apply
```

```
31 End With
```

```
32 ResetCalc
```

```
33 End With
```

```
34 End Sub
```

```
36 Sub Payroll_EmpToPay_SelectAll()
```

```
37 Dim LastSelRow As Long
```

```
38 With Sheet16
```

```
39 LastSelRow = .Range("L999").End(xlUp).Row
```

```
40 If LastSelRow < 11 Then Exit Sub
```

```
41 If .Range("B8").Value = True Then .Range("L11:L" & LastSelRow).Value = "p" Else: .Range("L11:L" & LastSelRow).Value = ""
```

```
42 End With
```

```
43 End Sub
```

```
45 Sub Payroll_FilterEmployees()
```

```
46 Dim LastRow As Long
```

```
47 Dim LastFiltRow As Long
```

```
48 Sheet2.Range("BF3:BI3,BF6:BG9999").ClearContents 'Clear Previous Criteria & Results
```

```
49 With Sheet16
```

```
50 .Range("D11:H9999").ClearContents 'Clear Existing Employee Results on Payroll Sheet
```

```
51 If .Range("H7").Value <> "[All Types]" Then Sheet2.Range("BF3").Value = .Range("H7").Value 'Employee Type Position Criteria
```

```
52 If .Range("F7").Value <> "[All Statuses]" Then Sheet2.Range("BG3").Value = .Range("F7").Value 'Employee Status Criteria
```

```
53 If .Range("F8").Value <> "[All Pay Types]" Then Sheet2.Range("BH3").Value = .Range("F8").Value 'Employee Pay Type Criteria
```

```
54 If .Range("H8").Value <> "[All Freq.]" Then Sheet2.Range("BI3").Value = .Range("H8").Value
```

```
1 2
```

```

1 2
) .Value 'Employee Type Salary Frequency Criteria
55 LastRow = Sheet2.Range("A999" ).End(xlUp).Row 'Last Employee Row
56 If .Range("B6" ).Value = False Then 'Do not Filter By Time Period
57     Sheet2.Range("A3:BC" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=Sheet2.
        Range("BF2:BI3" ), CopytoRange:=Sheet2.Range("BF5:BG5" ), Unique:=True
58 Else: 'Filter By Time Period
59     Sheet2.Range("A3:BD" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=Sheet2.
        Range("BE2:BI3" ), CopytoRange:=Sheet2.Range("BF5:BG5" ), Unique:=True
60 End If
61 LastFiltRow = Sheet2.Range("BF9999" ).End(xlUp).Row
62 If LastFiltRow < 6 Then GoTo NoData
63 If .Range("B6" ).Value = True Then ' Filter By Pay Period Time Clock Data
64
65
66 End If
67 .Range("E11:F" & LastFiltRow + 5).Value = Sheet2.Range("BF6:BG" & LastFiltRow).Value
    'Copy Over Employee Data
68 .Range("D11:D" & LastFiltRow + 5).Value = "p" 'Add Selected checkbox
69 NoData:
70 End With
71
72 End Sub
73
74 Sub Payroll_RemoveSelectedEmpl()
75     Dim PaidRow As Long
76     Dim SelRow As Long
77     Dim LastPaidRow As Long
78     Dim LastSelRow As Long
79     Dim FoundEmpIDRng As Range
80     With Sheet16
81         StopCalc
82         LastSelRow = .Range("D999" ).End(xlUp).Row
83         LastPaidRow = .Range("L999" ).End(xlUp).Row
84         If LastPaidRow < 10 Then LastPaidRow = 10
85         For SelRow = 11 To LastPaidRow
86             If .Range("L" & SelRow).Value = "p" Then
87                 'Check for Duplicates
88                 Set FoundEmpIDRng = .Range("E11:E" & LastSelRow).Find(.Range("M" & SelRow).
                    Value, , xlValues, xlWhole)
89                 If FoundEmpIDRng Is Nothing Then 'not Found
90                     LastSelRow = LastSelRow + 1
91                     .Range("D" & LastSelRow & ":F" & LastSelRow).Value = .Range("L" & SelRow &
                        ":N" & SelRow).Value 'Move Employee To First Available Row
92                     .Range("D" & LastSelRow).Value = "" 'Uncheck moved over selection
93                 End If
94                 .Range("L" & SelRow & ":N" & SelRow).ClearContents 'Clear Moved Over values
95             End If
96         Next SelRow
97         .Sort.SortFields.Clear
98         .Sort.SortFields.Add Key:=.Range("N11" ), SortOn:=xlSortOnValues, Order:=xlAscending,
            DataOption:=xlSortNormal
99         With Sheet16.Sort
100             .SetRange Range("L11:N" & LastPaidRow)
101             .Apply
102         End With
103         ResetCalc
104     End With
105 End Sub
106
107 Sub Payroll_SelToPay_SelectAll()
108     Dim LastSelRow As Long

```

```
109 1
110   With Sheet16
111     LastSelRow = .Range("D9999" ).End(xlUp).Row
112     If LastSelRow < 11 Then Exit Sub
113     If .Range("B7" ).Value = True Then .Range("D11:D" & LastSelRow).Value = "p" Else: .
114     Range("D11:D" & LastSelRow).Value = ""
115   End With
116 End Sub
```

**A**

Add, [139](#), [140](#)  
AdvancedFilter, [140](#)  
Apply, [139](#), [140](#)

**C**

Clear, [139](#), [140](#)  
ClearContents, [139](#), [140](#)  
CopytoRange, [140](#)  
CriteriaRange, [140](#)

**D**

DataOption, [139](#), [140](#)

**E**

Explicit, [139](#)

**F**

Find, [139](#), [140](#)  
FoundEmpIDRng, [139](#), [140](#)

**K**

Key, [139](#), [140](#)

**L**

LastFiltRow, [139](#), [140](#)  
LastPaidRow, [139](#), [140](#)  
LastRow, [139](#), [140](#)  
LastSelRow, [139-141](#)

**N**

NoData, [140](#)

**O**

Order, [139](#), [140](#)

**P**

PaidRow, [139](#), [140](#)  
Payroll\_AddSelectedEmpl, [139](#)  
Payroll\_EmpToPay\_SelectAll, [139](#)  
Payroll\_FilterEmployees, [139](#)  
Payroll\_RemoveSelectedEmpl, [140](#)  
Payroll\_SelToPay\_SelectAll, [140](#)

**R**

Range, [139-141](#)  
ResetCalc, [139](#), [140](#)  
Row, [139-141](#)

**S**

SelRow, [139](#), [140](#)  
SetRange, [139](#), [140](#)  
Sheet16, [139-141](#)  
Sheet2, [139](#), [140](#)  
Sort, [139](#), [140](#)  
SortFields, [139](#), [140](#)  
SortOn, [139](#), [140](#)  
StopCalc, [139](#), [140](#)

**U**

Unique, [140](#)

**V**

Value, [139-141](#)

**X**

xlAscending, [139](#), [140](#)  
xlFilterCopy, [140](#)  
xlSortNormal, [139](#), [140](#)

xlSortOnValues, [139](#), [140](#)

xlUp, [139-141](#)

xlValues, [139](#), [140](#)

xlWhole, [139](#), [140](#)

## Option Explicit

```
Sub Payroll_ConfirmCreate()
```

```
With Sheet16
```

```
Dim PayrollRow As Long
```

```
Dim LastEmpPayRow As Long
```

```
Dim LastEmpFiltRow As Long
```

```
If .Range("B3").Value = Empty Then 'Check for proper payroll
```

```
MsgBox "Please select a valid payroll"
```

```
.Range("BK3").Select
```

```
Exit Sub
```

```
End If
```

```
If .Range("BL3").Value = "Confirmed" Then 'Check for Previously confirmed
```

```
MsgBox "This payroll has already been confirmed"
```

```
Exit Sub
```

```
End If
```

```
If .Range("BL3").Value = "Pending Review" Then 'Check For Payroll yet to be reviewed
```

```
MsgBox "Please first review individual employees before marking this payroll as  
Confirmed"
```

```
Exit Sub
```

```
End If
```

```
PayrollRow = .Range("B3").Value 'Set Payroll Row
```

```
Sheet18.Range("F" & PayrollRow).Value = "Confirmed"
```

```
.Range("BL3").Value = "Confirmed"
```

```
'Confirmed Fade Out Message
```

```
'Update Shape Group
```

```
.Shapes("PaySummConfirmedGrp").Visible = msoCTrue
```

```
.Shapes("PaySummPendingGrp").Visible = msoFalse
```

```
End With
```

```
End Sub
```

```
Sub Payroll_LoadPayrollSummary()
```

```
With Sheet16
```

```
Dim PayrollRow As Long
```

```
Dim LastEmpPayRow As Long
```

```
Dim LastEmpFiltRow As Long
```

```
PayrollRow = .Range("B3").Value 'Set Payroll Row
```

```
.Range("BD6").Value = Sheet18.Range("C" & PayrollRow).Value 'Payroll Date
```

```
.Range("BG6").Value = Sheet18.Range("D" & PayrollRow).Value 'Payroll From
```

```
.Range("BI6").Value = Sheet18.Range("E" & PayrollRow).Value 'Payroll To
```

```
.Range("BL3").Value = Sheet18.Range("F" & PayrollRow).Value 'Payroll Status
```

```
.Range("BC9,BC10:BI500").ClearContents 'Clear Prev Payroll Data
```

```
Sheet20.Range("N3:Q9999").ClearContents 'Clear Prev Advanced Filter Results
```

```
Sheet20.Range("L3").Value = .Range("B2").Value 'Set Payroll ID for Criteria
```

```
LastEmpPayRow = Sheet20.Range("A9999").End(xlUp).Row 'Last Employee Pay Row
```

```
If LastEmpPayRow < 3 Then GoTo NoPayData
```

```
Sheet20.Range("A2:E" & LastEmpPayRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
```

```
Sheet20.Range("L2:L3"), CopytoRange:=Sheet20.Range("N2"), Unique:=True
```

```
LastEmpFiltRow = Sheet20.Range("N9999").End(xlUp).Row 'Last Filter Results Row
```

```
If LastEmpFiltRow < 3 Then GoTo NoPayData
```

```
.Range("BC9:BC" & LastEmpFiltRow + 6).Value = Sheet20.Range("N3:N" & LastEmpFiltRow).  
Value 'Copy Over Employee ID's
```

```
.Range("BD9:BI" & LastEmpFiltRow + 6).Formula = .Range("BD9:BI9").Formula 'Copy Down  
formulas
```

```
57 1 2 NoPayData:
58     End With
59 End Sub
60
61 Sub Payroll_PrintPayStubs()
62     Dim LastEmpRow As Long
63     Dim EmpRow As Long
64     If MsgBox("Are you sure you want to Print pay stubs for all employees?" , vbYesNo, "Print
        Pay Stubs" ) = vbNo Then Exit Sub
65     With Sheet16
66         If .Range("BL3" ).Value <> "Confirmed" Then 'Check for Previously confirmed
67             MsgBox "Please Confirm this Payroll before printing pay stubs"
68             Exit Sub
69         End If
70         LastEmpRow = .Range("BC9999" ).End(xlUp).Row
71         If LastEmpRow < 9 Then Exit Sub
72         Sheet15.Range("H4" ).ClearContents 'Clear any Existing Employee
73         Sheet15.Range("F3" ).Value = .Range("BK3" ).Value 'Set Payroll in Reports Screen
74         Sheet15.Activate
75         For EmpRow = 9 To LastEmpRow
76             Sheet15.Range("H4" ).Value = .Range("BD" & EmpRow).Value 'Set Employee & Run
77             Report
78             Reports_PrintOut
79         Next EmpRow
80     End With
End Sub
```

## A

Activate, [144](#)  
AdvancedFilter, [143](#)

## C

ClearContents, [143](#), [144](#)  
CopytoRange, [143](#)  
CriteriaRange, [143](#)

## E

EmpRow, [144](#)  
Empty, [143](#)  
Explicit, [143](#)

## F

Formula, [143](#)

## L

LastEmpFiltRow, [143](#)  
LastEmpPayRow, [143](#)  
LastEmpRow, [144](#)

## M

MsgBox, [143](#), [144](#)  
msoCTrue, [143](#)  
msoFalse, [143](#)

## N

NoPayData, [143](#), [144](#)

## P

Payroll\_ConfirmCreate, [143](#)  
Payroll\_LoadPayrollSummary, [143](#)  
Payroll\_PrintPayStubs, [144](#)  
PayrollRow, [143](#)

## R

Range, [143](#), [144](#)  
Reports\_PrintOut, [144](#)  
Row, [143](#), [144](#)

## S

Shapes, [143](#)  
Sheet15, [144](#)  
Sheet16, [143](#), [144](#)  
Sheet18, [143](#)  
Sheet20, [143](#)

## U

Unique, [143](#)

## V

Value, [143](#), [144](#)  
vbNo, [144](#)  
vbYesNo, [144](#)  
Visible, [143](#)

## X

xlFilterCopy, [143](#)  
xlUp, [143](#), [144](#)

```
1 Dim SelCell As Range
2 Dim DayName As String
3
4 .....Color Calendar Background'.....
5 Sub CalCol1()
6     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
7         With Selection.ShapeRange.Fill
8             .ForeColor.RGB = RGB(234, 234, 234)
9         End With
10        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
11    End With
12 End Sub
13 Sub CalCol2()
14     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
15         With Selection.ShapeRange.Fill
16             .ForeColor.RGB = RGB(197, 190, 151)
17         End With
18        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
19    End With
20 End Sub
21 Sub CalCol3()
22     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
23         With Selection.ShapeRange.Fill
24             .ForeColor.RGB = RGB(141, 180, 227)
25         End With
26        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
27    End With
28 End Sub
29 Sub CalCol4()
30     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
31         With Selection.ShapeRange.Fill
32             .ForeColor.RGB = RGB(184, 204, 228)
33         End With
34        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
35    End With
36 End Sub
37 Sub CalCol5()
38     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
39         With Selection.ShapeRange.Fill
40             .ForeColor.RGB = RGB(230, 185, 184)
41         End With
42        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
43    End With
44 End Sub
45 Sub CalCol6()
46     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
47         With Selection.ShapeRange.Fill
48             .ForeColor.RGB = RGB(215, 228, 188)
49         End With
50        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
51    End With
52 End Sub
53 Sub CalCol7()
54     With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
55         With Selection.ShapeRange.Fill
56             .ForeColor.RGB = RGB(204, 192, 218)
57         End With
58        ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
59    End With
60 End Sub
61 Sub CalCol8()
```

1



```

1
62 With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
63 With Selection.ShapeRange.Fill
64 .ForeColor.RGB = RGB(182, 221, 232)
65 End With
66 ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
67 End With
68 End Sub
69 Sub CalCol9()
70 With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
71 With Selection.ShapeRange.Fill
72 .ForeColor.RGB = RGB(252, 213, 180)
73 End With
74 ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
75 End With
76 End Sub
77 Sub CalendarHide()
78 Dim DayNum As Long
79 'Hide Calendar, Reset Day Colors
80 On Error GoTo NoCal
81 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
82 Sheets("CalPopUp" ).Range("A7" ).Value = ""
83 If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
84 For DayNum = 1 To 42
85 DayName = DayNum & "Day"
86 With ActiveSheet.Shapes(DayName)
87 .Fill.ForeColor.RGB = RGB(255, 255, 255)
88 .TextFrame2.TextRange.Font.Bold = msoFalse
89 End With
90 Next DayNum
91 End If
92 Exit Sub
93 NoCal: 'If calendar has been removed by accident, paste in backup calendar from CalPopUp
Sheet
94 End Sub
95
96 Sub CalendarShow()
97 With ActiveSheet
98 On Error Resume Next
99 Set SelCell = Selection
100 'Check if active cell is a valid date
101 If IsDate(SelCell.Value) = True Then
102 Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
103 Else: 'If No Date or incorrect Date user current date
104 Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
105 End If
106 Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).Value)
'Set Month
107 Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).Value)
'Set Year
108 DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
109 UngroupCal
110 If InStr(.Shapes("NextTri" ).OnAction, "!" ) <> 0 Or InStr(.Shapes("1Day" ).
DrawingObject.Formula, "]" ) <> 0 Then 'Run Workbook Link Remover and Cell Link
Replacement
111 MacroLinkRemover
112 CalFormulaReplacement
113 End If
114 GroupCal
115 On Error GoTo NoCal
116 With ActiveSheet.Shapes(DayName)
117 .Fill.ForeColor.RGB = RGB(252, 213, 180)

```

1 2 3

```

1 2 3
118 .TextFrame2.TextRange.Font.Bold = msoTrue
119 End With
120 On Error GoTo NoCal
121 .Shapes("Calendar").Visible = msoCTrue
122 .Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3", "CalCol4",
"CalCol5", "CalCol6", "CalCol7", "CalCol8", "CalCol9")).Visible = False '
123 .Shapes("Calendar").Left = SelCell.Left
124 .Shapes("Calendar").Top = SelCell.Offset(1, 0).Top
125 If Sheets("CalPopUp").Range("A6").Value > 0 Then
126 .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
"42Day")).Visible = True
127 Else:
128 .Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day", "41Day",
"42Day")).Visible = False
129 End If
130 Sheets("CalPopUp").Range("A7").Value = SelCell.Address
131 ActiveCell.Select
132 End With
133 Exit Sub
134 NoCal:
135 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the calendar
over from another sheet and paste into this sheet"
136 End Sub
137 Sub CalFormulaReplacement()
138 With ActiveSheet
139 Dim DayNum, ColNum, RowNum As Long
140 Dim shp As Shape
141 ColNum = 2
142 RowNum = 1
143 For DayNum = 1 To 42
144 .Shapes(DayNum & "Day").DrawingObject.Formula = "=CalPopUp!" & .Cells(RowNum,
ColNum).Address
145 ColNum = ColNum + 1
146 If ColNum = 9 Then
147 ColNum = 2
148 RowNum = RowNum + 1
149 End If
150 Next DayNum
151 .Shapes("MonthYear").DrawingObject.Formula = "=CalPopUp!$A$5"
152 End With
153 End Sub
154
155 Sub CheckForSheet()
156 'Checks for existence of Calendar Pop-up Worksheet
157 Dim ws As Worksheet
158 On Error GoTo CreateWS
159 Set ws = ActiveWorkbook.Sheets("CalPopUp")
160 Exit Sub
161 CreateWS:
162 CreateCalSht
163 End Sub
164
165 'Create Calendar Sheet on First Run of Calendar
166 Sub CreateCalSht()
167 Dim ColCnt, RowCnt, DayCnt, CalCol As Long
168 Dim ws, ActSht As Worksheet
169 Set ActSht = ActiveSheet
170 'On Error GoTo NoCal
171 ActiveSheet.Shapes("Calendar").Copy
172 Set ws = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))
173 ws.Name = "CalPopUp"

```

1

```

1
174 ActSht.Activate
175
176 'Reassign Shape Links & Macros
177 With ActiveSheet
178     UnGroupCal
179     '.Unprotect
180     .Shapes("NextRec" ).OnAction = "" & ActiveWorkbook.Name & "!NextMonth"
181     .Shapes("NextTri" ).OnAction = "" & ActiveWorkbook.Name & "!NextMonth"
182     .Shapes("PrevRec" ).OnAction = "" & ActiveWorkbook.Name & "!PrevMonth"
183     .Shapes("PrevTri" ).OnAction = "" & ActiveWorkbook.Name & "!PrevMonth"
184     .Shapes("SetBtn" ).OnAction = "" & ActiveWorkbook.Name & "!ShowSettings"
185     .Shapes("MonthYear" ).DrawingObject.Formula = "=CalPopUp!A5"
186     DayCnt = 1
187     For RowCnt = 1 To 6
188         For ColCnt = 2 To 8
189             .Shapes(DayCnt & "Day" ).DrawingObject.Formula = "=CalPopUp!" & .Cells(RowCnt,
190             ColCnt).Address 'Assigned Linked Cell
191             .Shapes(DayCnt & "Day" ).OnAction = "" & ActiveWorkbook.Name & "!DayBtn" &
192             DayCnt 'Assign Macro
193             DayCnt = DayCnt + 1
194         Next ColCnt
195     Next RowCnt
196
197 'Assign Color Macros
198 For CalCol = 1 To 9
199     .Shapes("CalCol" & CalCol).OnAction = "" & ActiveWorkbook.Name & "!CalCol" &
200     CalCol
201 Next CalCol
202 End With
203
204 With Sheets("CalPopUp" )
205     .Paste
206     .Visible = xlSheetHidden
207
208 'Add in Formulas and Details
209 .Range("A4" ).Value = "=INDEX(CalMonths,A3,)"
210 .Range("A5" ).Value = "=A4&" & Chr(34) & " " & Chr(34) & "&CalYear"
211 .Range("A6" ).Value = "=SUM(B6:H6)"
212 .Range("A8" ).Value = "January"
213 .Range("A8" ).AutoFill Destination:=.Range("A8:A19" ), Type:=xlFillDefault
214 .Range("A20" ).Value =
215 "=IFERROR(INDIRECT(ADDRESS(SUMPRODUCT((B1:H6=A1)*ROW(B1:H6))+6,SUMPRODUCT((B1:H6=A1)*COL
216 UMN(B1:H6)),1,1)),)" & Chr(34) & Chr(34) & ""
217
218 'Set Defined Names
219 ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
220 ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
221
222 'Add in Calendar Formulas
223 .Range("B1" ).Value =
224 "=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,CalMonth
225 s,0),1)," & Chr(34) & Chr(34) & ""
226 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
227 ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A$4,Cal
228 Months,0),1)," & Chr(34) & Chr(34) & ""
229 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
230 ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A$4,Cal
231 Months,0),1)," & Chr(34) & Chr(34) & ""
232 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
233 ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A$4,Cal
234 Months,0),1)," & Chr(34) & Chr(34) & ""
235 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &

```

1 2

```

",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A$4,Cal
Months,0),1)," & Chr(34) & Chr(34) & ")")
222 .Range("G1" ).Value = "=IF(F1<>" & Chr(34) & Chr(34) &
",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A$4,Cal
Months,0),1)," & Chr(34) & Chr(34) & ")")
223 .Range("H1" ).Value = "=IF(G1<>" & Chr(34) & Chr(34) &
",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A$4,Cal
Months,0),1)," & Chr(34) & Chr(34) & ")")
224 .Range("B2" ).Value = "=H1+1"
225 .Range("C2" ).Value = "=B2+1"
226 .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
227 .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
228 .Range("B5" ).Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3)," & Chr
(34) & Chr(34) & ",H4+1)"
229 .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) &
",B5+1)," & Chr(34) & Chr(34) & ")")
230 .Range("B6" ).Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) &
",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) & ")")
231 .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) & ",I5+1),"
& Chr(34) & Chr(34) & ")")
232 .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
233
234 'Set format to Single Day
235 .Range("B1:H6" ).NumberFormat = "d"
236
237 'Add in relative Day #'s
238 .Range("B7" ).Value = "1"
239 .Range("C7" ).Value = "2"
240 .Range("B8" ).Value = "8"
241 .Range("C8" ).Value = "9"
242 .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
243 .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
244 GroupCal
245 End With
246 Exit Sub
247 NoCal:
248 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the calendar
over from another sheet and paste into this sheet"
249 End Sub
250
251 .....
252 ''''Individual Day Selection Buttons
253 .....
254
255 Sub DayBtn1()
256 With Sheets("CalPopUp" )
257 If .Range("B1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
258 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B1" ).Value
259 End If
260 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
261 ActiveCell.Offset(0, 1).Select
262 End With
263 End Sub
264 Sub DayBtn10()
265 With Sheets("CalPopUp" )
266 If .Range("D2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
267 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D2" ).Value
268 End If
269 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
270 ActiveCell.Offset(0, 1).Select
271 End With

```

1

```
1
272 End Sub
273 Sub DayBtn11()
274     With Sheets("CalPopUp" )
275         If .Range("E2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
276             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E2" ).Value
277         End If
278         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
279         ActiveCell.Offset(0, 1).Select
280     End With
281 End Sub
282 Sub DayBtn12()
283     With Sheets("CalPopUp" )
284         If .Range("F2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
285             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F2" ).Value
286         End If
287         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
288         ActiveCell.Offset(0, 1).Select
289     End With
290 End Sub
291 Sub DayBtn13()
292     With Sheets("CalPopUp" )
293         If .Range("G2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
294             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G2" ).Value
295         End If
296         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
297         ActiveCell.Offset(0, 1).Select
298     End With
299 End Sub
300 Sub DayBtn14()
301     With Sheets("CalPopUp" )
302         If .Range("H2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
303             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H2" ).Value
304         End If
305         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
306         ActiveCell.Offset(0, 1).Select
307     End With
308 End Sub
309 Sub DayBtn15()
310     With Sheets("CalPopUp" )
311         If .Range("B3" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
312             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B3" ).Value
313         End If
314         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
315         ActiveCell.Offset(0, 1).Select
316     End With
317 End Sub
318 Sub DayBtn16()
319     With Sheets("CalPopUp" )
320         If .Range("C3" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
321             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C3" ).Value
322         End If
323         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
324         ActiveCell.Offset(0, 1).Select
325     End With
326 End Sub
327 Sub DayBtn17()
328     With Sheets("CalPopUp" )
329         If .Range("D3" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
330             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D3" ).Value
331         End If
332         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse

```

1 2

```
1 2
333     ActiveCell.Offset(0, 1).Select
334     End With
335 End Sub
336 Sub DayBtn18()
337     With Sheets("CalPopUp" )
338         If .Range("E3" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
339             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E3" ).Value
340         End If
341         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
342         ActiveCell.Offset(0, 1).Select
343     End With
344 End Sub
345 Sub DayBtn19()
346     With Sheets("CalPopUp" )
347         If .Range("F3" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
348             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F3" ).Value
349         End If
350         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
351         ActiveCell.Offset(0, 1).Select
352     End With
353 End Sub
354 Sub DayBtn2()
355     With Sheets("CalPopUp" )
356         If .Range("C1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
357             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C1" ).Value
358         End If
359         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
360         ActiveCell.Offset(0, 1).Select
361     End With
362 End Sub
363 Sub DayBtn20()
364     With Sheets("CalPopUp" )
365         If .Range("G3" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
366             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G3" ).Value
367         End If
368         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
369         ActiveCell.Offset(0, 1).Select
370     End With
371 End Sub
372 Sub DayBtn21()
373     With Sheets("CalPopUp" )
374         If .Range("H3" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
375             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H3" ).Value
376         End If
377         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
378         ActiveCell.Offset(0, 1).Select
379     End With
380 End Sub
381 Sub DayBtn22()
382     With Sheets("CalPopUp" )
383         If .Range("B4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
384             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B4" ).Value
385         End If
386         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
387         ActiveCell.Offset(0, 1).Select
388     End With
389 End Sub
390 Sub DayBtn23()
391     With Sheets("CalPopUp" )
392         If .Range("C4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
393             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C4" ).Value
```

1 2 3

```
1 2 3
394     End If
395     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
396     ActiveCell.Offset(0, 1).Select
397 End With
398 End Sub
399 Sub DayBtn24()
400     With Sheets("CalPopUp" )
401         If .Range("D4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
402             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D4" ).Value
403         End If
404         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
405         ActiveCell.Offset(0, 1).Select
406     End With
407 End Sub
408 Sub DayBtn25()
409     With Sheets("CalPopUp" )
410         If .Range("E4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
411             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E4" ).Value
412         End If
413         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
414         ActiveCell.Offset(0, 1).Select
415     End With
416 End Sub
417 Sub DayBtn26()
418     With Sheets("CalPopUp" )
419         If .Range("F4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
420             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F4" ).Value
421         End If
422         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
423         ActiveCell.Offset(0, 1).Select
424     End With
425 End Sub
426 Sub DayBtn27()
427     With Sheets("CalPopUp" )
428         If .Range("G4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
429             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G4" ).Value
430         End If
431         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
432         ActiveCell.Offset(0, 1).Select
433     End With
434 End Sub
435 Sub DayBtn28()
436     With Sheets("CalPopUp" )
437         If .Range("H4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
438             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H4" ).Value
439         End If
440         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
441         ActiveCell.Offset(0, 1).Select
442     End With
443 End Sub
444 Sub DayBtn29()
445     With Sheets("CalPopUp" )
446         If .Range("B5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
447             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B5" ).Value
448         End If
449         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
450         ActiveCell.Offset(0, 1).Select
451     End With
452 End Sub
453 Sub DayBtn3()
454     With Sheets("CalPopUp" )
```

1 2



```
1 2
455 If .Range("D1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
456     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D1" ).Value
457 End If
458     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
459     ActiveCell.Offset(0, 1).Select
460 End With
461 End Sub
462 Sub DayBtn30()
463     With Sheets("CalPopUp" )
464         If .Range("C5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
465             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C5" ).Value
466         End If
467         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
468         ActiveCell.Offset(0, 1).Select
469     End With
470 End Sub
471 Sub DayBtn31()
472     With Sheets("CalPopUp" )
473         If .Range("D5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
474             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D5" ).Value
475         End If
476         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
477         ActiveCell.Offset(0, 1).Select
478     End With
479 End Sub
480 Sub DayBtn32()
481     With Sheets("CalPopUp" )
482         If .Range("E5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
483             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E5" ).Value
484         End If
485         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
486         ActiveCell.Offset(0, 1).Select
487     End With
488 End Sub
489 Sub DayBtn33()
490     With Sheets("CalPopUp" )
491         If .Range("F5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
492             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F5" ).Value
493         End If
494         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
495         ActiveCell.Offset(0, 1).Select
496     End With
497 End Sub
498 Sub DayBtn34()
499     With Sheets("CalPopUp" )
500         If .Range("G5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
501             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G5" ).Value
502         End If
503         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
504         ActiveCell.Offset(0, 1).Select
505     End With
506 End Sub
507 Sub DayBtn35()
508     With Sheets("CalPopUp" )
509         If .Range("H5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
510             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H5" ).Value
511         End If
512         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
513         ActiveCell.Offset(0, 1).Select
514     End With
515 End Sub
```



```
516 Sub DayBtn36()  
517 With Sheets("CalPopUp" )  
518 If .Range("B6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then  
519 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B6" ).Value  
520 End If  
521 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse  
522 ActiveCell.Offset(0, 1).Select  
523 End With  
524 End Sub  
525 Sub DayBtn37()  
526 With Sheets("CalPopUp" )  
527 If .Range("C6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then  
528 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C6" ).Value  
529 End If  
530 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse  
531 ActiveCell.Offset(0, 1).Select  
532 End With  
533 End Sub  
534 Sub DayBtn38()  
535 With Sheets("CalPopUp" )  
536 If .Range("D6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then  
537 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D6" ).Value  
538 End If  
539 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse  
540 ActiveCell.Offset(0, 1).Select  
541 End With  
542 End Sub  
543 Sub DayBtn39()  
544 With Sheets("CalPopUp" )  
545 If .Range("E6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then  
546 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E6" ).Value  
547 End If  
548 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse  
549 ActiveCell.Offset(0, 1).Select  
550 End With  
551 End Sub  
552 Sub DayBtn4()  
553 With Sheets("CalPopUp" )  
554 If .Range("E1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then  
555 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E1" ).Value  
556 End If  
557 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse  
558 ActiveCell.Offset(0, 1).Select  
559 End With  
560 End Sub  
561 Sub DayBtn40()  
562 With Sheets("CalPopUp" )  
563 If .Range("F6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then  
564 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F6" ).Value  
565 End If  
566 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse  
567 ActiveCell.Offset(0, 1).Select  
568 End With  
569 End Sub  
570 Sub DayBtn41()  
571 With Sheets("CalPopUp" )  
572 If .Range("G6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then  
573 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G6" ).Value  
574 End If  
575 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse  
576 ActiveCell.Offset(0, 1).Select
```

```
1 2
577 End With
578 End Sub
579 Sub DayBtn42()
580 With Sheets("CalPopUp" )
581 If .Range("H6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
582     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H6" ).Value
583 End If
584 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
585 ActiveCell.Offset(0, 1).Select
586 End With
587 End Sub
588 Sub DayBtn5()
589 With Sheets("CalPopUp" )
590 If .Range("F1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
591     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F1" ).Value
592 End If
593 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
594 ActiveCell.Offset(0, 1).Select
595 End With
596 End Sub
597 Sub DayBtn6()
598 With Sheets("CalPopUp" )
599 If .Range("G1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
600     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G1" ).Value
601 End If
602 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
603 ActiveCell.Offset(0, 1).Select
604 End With
605 End Sub
606 Sub DayBtn7()
607 With Sheets("CalPopUp" )
608 If .Range("H1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
609     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H1" ).Value
610 End If
611 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
612 ActiveCell.Offset(0, 1).Select
613 End With
614 End Sub
615 Sub DayBtn8()
616 With Sheets("CalPopUp" )
617 If .Range("B2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
618     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B2" ).Value
619 End If
620 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
621 ActiveCell.Offset(0, 1).Select
622 End With
623 End Sub
624 Sub DayBtn9()
625 With Sheets("CalPopUp" )
626 If .Range("C2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
627     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C2" ).Value
628 End If
629 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
630 ActiveCell.Offset(0, 1).Select
631 End With
632 End Sub
633 Sub GroupCal()
634 ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
635 Selection.ShapeRange.Name = "NextMonth"
636 ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
637 Selection.ShapeRange.Name = "PrevMonth"
```

1

```

1
638 ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
639     , "42Day" , "37Day" , "36Day" , "CalBack" , "MonthYear" , "CalBorder" , "1Day" , _
640     "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
641     "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
    "18Day" _
642     , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
643     "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" , "Sa" ,
    "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
    "CalCol3" , _
645     "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth" ,
    "NextMonth" )).Visible = msoCTrue
647 ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
648     , "42Day" , "37Day" , "36Day" , "CalBack" , "MonthYear" , "CalBorder" , "1Day" , _
649     "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
650     "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" , "18Day" _
651     , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
652     "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" , "Sa" , _
653     "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" , "CalCol3" ,
    "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth" ,
    "NextMonth" )).Select
656 Selection.ShapeRange.Group.Select
657 Selection.ShapeRange.Name = "Calendar"
658 Selection.Name = "Calendar"
659 Selection.Placement = xlMove
660 ActiveSheet.Shapes("Calendar" ).Placement = 2
661 End Sub

662
663
664
665
666
667
668
669
670 Sub MacroLinkRemover()
671     'PURPOSE: Remove an external workbook reference from all shapes triggering macros
672     'Source: www.ExcelForFreelancers.com
673     Dim shp As Shape
674     Dim MacroLink, NewLink As String
675     Dim SplitLink As Variant
676
677     For Each shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
678
679         'Grab current macro link (if available)
680         On Error GoTo NextShp
681         MacroLink = shp.OnAction
682
683         'Determine if shape was linking to a macro
684         If MacroLink <> "" And InStr(MacroLink, "!") <> 0 Then
685             'Split Macro Link at the exclamation mark (store in Array)
686             SplitLink = Split(MacroLink, "!")
687
688             'Pull text occurring after exclamation mark
689             NewLink = SplitLink(1)
690
691             'Remove any straggling apostrophes from workbook name
692             If Right(NewLink, 1) = "'" Then
693                 NewLink = Left(NewLink, Len(NewLink) - 1)

```

1 2 3 4

```

1 2 3 4
694     End If
695
696     'Apply New Link
697     shp.OnAction = NewLink
698     End If
699     NextShp:
700     Next shp
701 End Sub
702 Sub NextMonth()
703     'Next Month button
704     If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
705         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
706         With ActiveSheet.Shapes(DayName)
707             .Fill.ForeColor.RGB = RGB(255, 255, 255)
708             .TextFrame2.TextRange.Font.Bold = msoFalse
709         End With
710     End If
711     With Sheets("CalPopUp" )
712         If .Range("A3" ).Value = 12 Then
713             .Range("A3" ).Value = 1
714             .Range("A2" ).Value = .Range("A2" ).Value + 1
715         Else:
716             .Range("A3" ).Value = .Range("A3" ).Value + 1
717         End If
718         If .Range("A6" ).Value > 0 Then
719             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" ,
720             "41Day" , "42Day" )).Visible = True
721         Else:
722             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" ,
723             "41Day" , "42Day" )).Visible = False
724         End If
725     End With
726 End Sub
727 Sub PrevMonth()
728     'Previous Month Button
729     If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
730         DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
731         With ActiveSheet.Shapes(DayName)
732             .Fill.ForeColor.RGB = RGB(255, 255, 255)
733             .TextFrame2.TextRange.Font.Bold = msoFalse
734         End With
735     End If
736     With Sheets("CalPopUp" )
737         If .Range("A3" ).Value = 1 Then
738             .Range("A3" ).Value = 12
739             .Range("A2" ).Value = .Range("A2" ).Value - 1
740         Else:
741             .Range("A3" ).Value = .Range("A3" ).Value - 1
742         End If
743         If .Range("A6" ).Value > 0 Then
744             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" ,
745             "41Day" , "42Day" )).Visible = True
746         Else:
747             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" ,
748             "41Day" , "42Day" )).Visible = False
749         End If
750     End With
751 End Sub
752 Sub ShowSettings()
753     'Show or Hide Calendar Settings Panel
754     If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then

```

1 2

```
751 1 2 ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,  
"CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible =  
False  
752 } Else:  
753 ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,  
"CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible = True  
754 End If  
755 End Sub  
756  
757 Sub UnGroupCal()  
758 On Error Resume Next  
759 ActiveSheet.Shapes("Calendar" ).Ungroup  
760 ActiveSheet.Shapes("NextMonth" ).Ungroup  
761 ActiveSheet.Shapes("PrevMonth" ).Ungroup  
762 On Error GoTo 0  
763 End Sub
```

—, 157

## A

Activate, 149  
 ActiveCell, 148, 150-156  
 ActiveSheet, 146-159  
 ActiveWorkbook, 148, 149  
 ActSht, 148, 149  
 Add, 148, 149  
 Address, 148, 149  
 After, 148  
 Array, 146-148, 156-159  
 AutoFill, 149, 150

## B

Bold, 147, 148, 158

## C

CalCol, 148, 149  
 CalCol1, 146  
 CalCol2, 146  
 CalCol3, 146  
 CalCol4, 146  
 CalCol5, 146  
 CalCol6, 146  
 CalCol7, 146  
 CalCol8, 146  
 CalCol9, 147  
 CalendarHide, 147  
 CalendarShow, 147  
 CalFormulaReplacement, 147, 148  
 Cells, 148, 149  
 CheckForSheet, 148  
 Chr, 149, 150  
 ColCnt, 148, 149  
 ColNum, 148  
 Copy, 148  
 Count, 148  
 CreateCalSht, 148  
 CreateWS, 148

## D

DayBtn1, 150  
 DayBtn10, 150  
 DayBtn11, 151  
 DayBtn12, 151  
 DayBtn13, 151  
 DayBtn14, 151  
 DayBtn15, 151  
 DayBtn16, 151  
 DayBtn17, 151  
 DayBtn18, 152  
 DayBtn19, 152  
 DayBtn2, 152  
 DayBtn20, 152  
 DayBtn21, 152  
 DayBtn22, 152  
 DayBtn23, 152  
 DayBtn24, 153  
 DayBtn25, 153  
 DayBtn26, 153  
 DayBtn27, 153  
 DayBtn28, 153  
 DayBtn29, 153  
 DayBtn3, 153  
 DayBtn30, 154  
 DayBtn31, 154  
 DayBtn32, 154

DayBtn33, 154  
 DayBtn34, 154  
 DayBtn35, 154  
 DayBtn36, 155  
 DayBtn37, 155  
 DayBtn38, 155  
 DayBtn39, 155  
 DayBtn4, 155  
 DayBtn40, 155  
 DayBtn41, 155  
 DayBtn42, 156  
 DayBtn5, 156  
 DayBtn6, 156  
 DayBtn7, 156  
 DayBtn8, 156  
 DayBtn9, 156  
 DayCnt, 148, 149  
 DayName, 146, 147, 158  
 DayNum, 147, 148  
 Destination, 149, 150  
 DrawingObject, 147-149

## E

Empty, 147, 150-156, 158

## F

Fill, 146, 147, 158  
 Font, 147, 148, 158  
 ForeColor, 146, 147, 158  
 Formula, 147-149

## G

Group, 156, 157  
 GroupCal, 147, 150, 156

## I

InStr, 147, 157  
 IsDate, 147

## L

Left, 148, 157  
 Len, 157

## M

MacroLink, 157  
 MacroLinkRemover, 147, 157  
 Month, 147  
 MsgBox, 148, 150  
 msoCTrue, 148, 157  
 msoFalse, 147, 150-156, 158  
 msoTrue, 148

## N

Name, 148, 149, 156, 157  
 Names, 149  
 NewLink, 157, 158  
 NextMonth, 158  
 NextShp, 157, 158  
 NoCal, 147, 148, 150  
 NumberFormat, 150

## O

Offset, 148, 150-156  
 OnAction, 147, 149, 157, 158

## P

Paste, 149  
 Placement, 157  
 PrevMonth, 158

## R

Range, 146-159  
 RefersTo, 149  
 RGB, 146, 147, 158  
 Right, 157  
 RowCnt, 148, 149  
 RowNum, 148

## S

SelCell, 146-148  
 Selection, 146, 147, 156, 157  
 Shape, 148, 157  
 ShapeRange, 146, 147, 156, 157  
 Shapes, 146-159  
 Sheets, 146-156, 158  
 ShowSettings, 158  
 shp, 148, 157, 158  
 Split, 157  
 SplitLink, 157

## T

TextFrame2, 147, 148, 158  
 TextRange, 147, 148, 158  
 ThisWorkbook, 148  
 Top, 148

## U

Ungroup, 159  
 UngroupCal, 147, 149, 159

## V

Value, 146-156, 158  
 Visible, 147-159

## W

Worksheet, 148  
 ws, 148

## X

xlFillDefault, 149, 150  
 xlMove, 157  
 xlSheetHidden, 149

## Y

Year, 147

```
1 Option Explicit
2
3 Sub Reports_Clear()
4     With Sheet15
5         Dim ClearRng As Range
6         Set ClearRng = .Range("E5:Z99999" )
7         With ClearRng
8             .Clear
9             .ClearFormats
10            .UnMerge
11            .Locked = False 'Unlock cells in Range
12        End With
13    End With
14 End Sub
15
16 Sub Reports_PrintOut()
17     With Sheet15
18         Dim PrntRng As Range
19         Dim LastRow As Long
20         LastRow = .Range("F9999" ).End(xlUp).Row
21         Set PrntRng = .Range("F5:J" & LastRow)
22         .PageSetup.PrintArea = PrntRng.Address
23         .PrintOut , , 1, False, , , , False
24     End With
25 End Sub
```

**A**

Address, [161](#)

**C**

Clear, [161](#)

ClearFormats, [161](#)

ClearRng, [161](#)

**E**

Explicit, [161](#)

**L**

LastRow, [161](#)

Locked, [161](#)

**P**

PageSetup, [161](#)

PrintArea, [161](#)

PrintOut, [161](#)

PrntRng, [161](#)

**R**

Range, [161](#)

Reports\_Clear, [161](#)

Reports\_PrintOut, [161](#)

Row, [161](#)

**S**

Sheet15, [161](#)

**U**

UnMerge, [161](#)

**X**

xUp, [161](#)



```

1  Option Explicit
2
3  Sub Reports_EarningsStatement()
4      Dim PayrollRow As Long
5      Dim PayrollItemRow As Long
6      Dim LastPayrollItemRow As Long
7      Dim LastPayrollItemFiltRow As Long
8      Dim TimeInteger As Long
9      Dim ReportRow As Long
10     Dim LastReportRow As Long
11     Dim PayID As String
12     Dim EmpID As String
13     Dim PayType As String
14     Dim RptRng As Range
15     With Sheet15
16         If .Range("B6").Value = Empty Then Exit Sub 'Exit on No/Missing Payroll ID
17         If .Range("B9").Value = Empty Then Exit Sub 'Exit on missing Employee ID
18         StopCalc
19         .Range("AA17:AB17,AD17").ClearContents
20         .Range("AA18:AE9999").Clear 'Clear All Conents and Formats
21         TimeInteger = Sheet4.Range("TimeInteger").Value 'Set Time Integer for Decimal/hourly
22         PayID = .Range("B6").Value 'Set Payroll ID
23         EmpID = .Range("B9").Value 'Set Employee ID
24         PayrollRow = .Range("B4").Value 'Set Payroll Row
25         .Range("B3").Value = 1 'Set Report # to 1
26         .Range("B2").Value = True 'Set Load Report to True
27         Sheet19.Range("M3:V99999").ClearContents 'Clear Payroll Item List Criteria & Results
28         Sheet19.Range("M3").Value = PayID 'Payroll ID
29         Sheet19.Range("N3").Value = EmpID 'Employee ID
30         LastPayrollItemRow = Sheet19.Range("A99999").End(xlUp).Row 'Last Payroll Item Row
31         Sheet19.Range("A2:J" & LastPayrollItemRow).AdvancedFilter xlFilterCopy, CriteriaRange:=Sheet19.Range("M2:N3"), CopytoRange:=Sheet19.Range("P2:V2"), Unique:=True
32         LastPayrollItemFiltRow = Sheet19.Range("P999999").End(xlUp).Row
33         If LastPayrollItemFiltRow < 3 Then GoTo NoPayData
34         ReportRow = 17 'Set Intial Report Row
35
36         'Check For Regular Pay OR Salary Income Items - First loop- Regular & Salary Pay
37         For PayrollItemRow = 3 To LastPayrollItemRow
38             PayType = Sheet19.Range("P" & PayrollItemRow).Value 'Payroll Item Type
39             If PayType = "Regular Pay" Or PayType = "Salary Pay" Then
40                 .Range("AA" & ReportRow).Value = Sheet19.Range("Q" & PayrollItemRow).Value
41                 'Item Name
42                 On Error Resume Next
43                 .Range("AB" & ReportRow).Value = 1 / (TimeInteger / Sheet19.Range("S" & PayrollItemRow).Value) 'Hours
44                 On Error GoTo 0
45                 .Range("AD" & ReportRow).Value = Sheet19.Range("R" & PayrollItemRow).Value
46                 'Amount
47                 'Copy Formulas
48                 .Range("AA" & ReportRow & ":AE" & ReportRow).Copy
49                 ReportRow = ReportRow + 1
50                 .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormats
51                 .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormulas
52                 .Range("AA" & ReportRow & ":AB" & ReportRow).ClearContents
53                 .Range("AD" & ReportRow).ClearContents
54             End If
55         Next PayrollItemRow
56
57         'Add in Overtime (if any) - 2nd Loop
58         For PayrollItemRow = 3 To LastPayrollItemRow

```

1 2 3

```

1 2 3
57 PayType = Sheet19.Range("P" & PayrollItemRow).Value 'Payroll Item Type
58 If PayType = "OT Pay" Then
59     .Range("AA" & ReportRow).Value = Sheet19.Range("Q" & PayrollItemRow).Value
        'Item Name
60     On Error Resume Next
61     .Range("AB" & ReportRow).Value = 1 / (TimeInteger / Sheet19.Range("S" &
        PayrollItemRow).Value) 'Hours
62     On Error GoTo 0
63     .Range("AD" & ReportRow).Value = Sheet19.Range("R" & PayrollItemRow).Value
        'Amount
64
65     'Copy Formulas & Formats
66     .Range("AA" & ReportRow & ":AE" & ReportRow).Copy
67     ReportRow = ReportRow + 1
68     .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormats
69     .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormulas
70     .Range("AA" & ReportRow & ":AB" & ReportRow).ClearContents
71     .Range("AD" & ReportRow).ClearContents
72 End If
73 Next PayrollItemRow
74
75 ReportRow = ReportRow + 1
76
77 'Add In Fed & Local Tax Deductions - 3rd Loop
78 .Range("RptRng_FedTax" ).Copy
79 .Range("AA" & ReportRow).PasteSpecial xlPasteAll
80 ReportRow = ReportRow + 2
81 For PayrollItemRow = 3 To LastPayrollItemRow
82     PayType = Sheet19.Range("P" & PayrollItemRow).Value 'Payroll Item Type
83     If PayType = "Tax Item" Then
84         .Range("AA" & ReportRow).Value = Sheet19.Range("Q" & PayrollItemRow).Value
            'Item Name
85         .Range("AD" & ReportRow).Value = Sheet19.Range("R" & PayrollItemRow).Value
            'Amount
86         'Copy Formulas & Formats
87         .Range("AA" & ReportRow & ":AE" & ReportRow).Copy
88         ReportRow = ReportRow + 1
89         .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormats
90         .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormulas
91         .Range("AA" & ReportRow & ":AD" & ReportRow).ClearContents
92     End If
93 Next PayrollItemRow
94
95
96 ReportRow = ReportRow + 1
97
98 'Add In Other Deductions- 4th Loop
99 .Range("RptRng_OthDed" ).Copy
100 .Range("AA" & ReportRow).PasteSpecial xlPasteAll
101 ReportRow = ReportRow + 2
102 For PayrollItemRow = 3 To LastPayrollItemRow
103     PayType = Sheet19.Range("P" & PayrollItemRow).Value 'Payroll Item Type
104     If PayType = "Other Deduction" Then
105         .Range("AA" & ReportRow).Value = Sheet19.Range("Q" & PayrollItemRow).Value
            'Item Name
106         .Range("AD" & ReportRow).Value = Sheet19.Range("R" & PayrollItemRow).Value
            'Amount
107         'Copy Formulas & Formats
108         .Range("AA" & ReportRow & ":AE" & ReportRow).Copy
109         ReportRow = ReportRow + 1
110         .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormats

```

1 2 3 4

```

1 2 3 4
111 .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormulas
112 .Range("AA" & ReportRow & ":AD" & ReportRow).ClearContents
113     End If
114 Next PayrollItemRow
115
116
117 ReportRow = ReportRow + 1
118
119 'Add In Other Additions 5th Loop
120 .Range("RptRng_OthAdd" ).Copy
121 .Range("AA" & ReportRow).PasteSpecial xlPasteAll
122 ReportRow = ReportRow + 2
123 For PayrollItemRow = 3 To LastPayrollItemRow
124     PayType = Sheet19.Range("P" & PayrollItemRow).Value 'Payroll Item Type
125     If PayType = "Other Addition" Then
126         .Range("AA" & ReportRow).Value = Sheet19.Range("Q" & PayrollItemRow).Value
127         .Range("AD" & ReportRow).Value = Sheet19.Range("R" & PayrollItemRow).Value
128         'Item Name
129         'Amount
130         'Copy Formulas & Formats
131         .Range("AA" & ReportRow & ":AE" & ReportRow).Copy
132         ReportRow = ReportRow + 1
133         .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormulas
134         .Range("AA" & ReportRow & ":AD" & ReportRow).ClearContents
135     End If
136 Next PayrollItemRow
137 ReportRow = ReportRow + 1
138
139
140 'Add Earnings Summary Range
141 .Range("RepRng_EarnSum" ).Copy
142 .Range("AA" & ReportRow).PasteSpecial xlPasteAll
143 Application.CutCopyMode = False
144 LastReportRow = ReportRow + 4 'Set Last Report Row
145 NoPayData:
146
147 If LastReportRow = 0 Then LastReportRow = 9 'In case of No Pay Data, set the Last
148 Report Row to 9 (we can't have this at 0)
149 Set RptRng = .Range("AA5:AE" & LastReportRow) 'Set the Range for the report
150 RptRng.Copy 'Copy the Report Range
151 With .Range("F5" )
152     .PasteSpecial xlPasteAll 'Paste Entire Report in Cell F5
153     .PasteSpecial xlPasteColumnWidths 'Paste all column widths of the original report
154     also in F5
155 End With
156 .Range("B2" ).Value = False 'Set Load Report to False
157 .Range("F3" ).Select
158 End With
159 ResetCalc
160 End Sub

```

**A**

AdvancedFilter, 163  
Application, 165

**C**

Clear, 163  
ClearContents, 163-165  
Copy, 163-165  
CopytoRange, 163  
CriteriaRange, 163  
CutCopyMode, 165

**E**

EmplID, 163  
Empty, 163  
Explicit, 163

**L**

LastPayrollItemFiltRow, 163  
LastPayrollItemRow, 163-165  
LastReportRow, 163, 165

**N**

NoPayData, 163, 165

**P**

PasteSpecial, 163-165  
PayID, 163  
PayrollItemRow, 163-165  
PayrollRow, 163  
PayType, 163-165

**R**

Range, 163-165  
ReportRow, 163-165  
Reports\_EarningsStatement, 163  
ResetCalc, 165  
Row, 163  
RptRng, 163, 165

**S**

Sheet15, 163  
Sheet19, 163-165  
Sheet4, 163  
StopCalc, 163

**T**

TimeInteger, 163, 164

**U**

Unique, 163

**V**

Value, 163-165

**X**

xlFilterCopy, 163  
xlPasteAll, 164, 165  
xlPasteColumnWidths, 165  
xlPasteFormats, 163-165  
xlPasteFormulas, 163-165  
xlUp, 163

```

1 Option Explicit
2
3 Sub Report_TimeclockReport()
4     Dim LastReportRow As Long
5     Dim LastTimeclockRow As Long
6     Dim LastFiltRow As Long
7     Dim RptRng As Range
8     StopCalc
9     With Sheet15
10        .Range("B3" ).Value = 2 'Set Report #
11        .Range("BA7:BH9999" ).ClearContents 'Clear Old Values
12        Sheet9.Range("Z3:AI3,AA5:AI9999" ).ClearContents
13        If .Range("H3" ).Value <> Empty Then Sheet9.Range("Z3" ).Value = ">=" & .Range("H3" ).
            Value 'From Date
14        If .Range("J3" ).Value <> Empty Then Sheet9.Range("AD3" ).Value = "<=" & .Range("J3" )
            .Value 'To Date
15        If .Range("B9" ).Value <> Empty Then Sheet9.Range("AB3" ).Value = .Range("B9" ).Value
            'Employee ID
16
17        LastTimeclockRow = Sheet9.Range("A99999" ).End(xlUp).Row 'Last TimeClock Row
18        Sheet9.Range("A4:H" & LastTimeclockRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
            Sheet9.Range("Z2:AD3" ), CopytoRange:=Sheet9.Range("AB4:AF4" ), Unique:=True
19        LastFiltRow = Sheet9.Range("AB9999" ).End(xlUp).Row
20        If LastFiltRow < 5 Then GoTo NoTimeData
21        Sheet9.Range("AG1:AI1" ).Copy
22        Sheet9.Range("AG5:AI" & LastFiltRow).PasteSpecial xlPasteFormulas 'Copy Total Hour
            Formulas
23        Sheet9.Calculate
24        .Range("BA7:BH" & LastFiltRow + 2).Value = Sheet9.Range("AB5:AI" & LastFiltRow).Value
            'Copy Report Data
25
26        LastReportRow = LastFiltRow + 2
27        NoTimeData:
28        If LastReportRow < 8 Then LastReportRow = 8 'In case of No Data or just 1 row of data
            set the Last Report Row to 8 to include summary
29
30        Set RptRng = .Range("BA5:BK" & LastReportRow) 'Set the Range for the report
31        RptRng.Copy 'Copy the Report Range
32        With .Range("F5" )
33            .PasteSpecial xlPasteAll 'Paste Entire Report in Cell F5
34            .PasteSpecial xlPasteColumnWidths 'Paste all column widths of the original report
                also in F5
35        End With
36
37        Application.CutCopyMode = False
38        .Range("B2" ).Value = False 'Set Load Report to False
39        .Range("F3" ).Select
40    End With
41    ResetCalc
42
43 End Sub

```

## A

AdvancedFilter, [167](#)  
Application, [167](#)

## C

Calculate, [167](#)  
ClearContents, [167](#)  
Copy, [167](#)  
CopytoRange, [167](#)  
CriteriaRange, [167](#)  
CutCopyMode, [167](#)

## E

Empty, [167](#)  
Explicit, [167](#)

## L

LastFilterRow, [167](#)  
LastReportRow, [167](#)  
LastTimeclockRow, [167](#)

## N

NoTimeData, [167](#)

## P

PasteSpecial, [167](#)

## R

Range, [167](#)  
Report\_TimelockReport, [167](#)  
ResetCalc, [167](#)  
Row, [167](#)  
RptRng, [167](#)

## S

Sheet15, [167](#)  
Sheet9, [167](#)  
StopCalc, [167](#)

## U

Unique, [167](#)

## V

Value, [167](#)

## X

xlFilterCopy, [167](#)  
xlPasteAll, [167](#)  
xlPasteColumnWidths, [167](#)  
xlPasteFormulas, [167](#)  
xlUp, [167](#)

```
1 Option Explicit
2
3 Sub GoTo_Admin(control As IRibbonControl)
4     Sheet4.Activate
5 End Sub
6 Sub GoTo_EmpArchive(control As IRibbonControl)
7     Sheet6.Activate
8 End Sub
9 Sub GoTo_EmpAttach(control As IRibbonControl)
10    Sheet3.Activate
11 End Sub
12 Sub GoTo_EmployeeID(control As IRibbonControl)
13    Sheet1.Activate
14    EmpIDCard_Show
15 End Sub
16
17 Sub GoTo_Employees(control As IRibbonControl)
18    Sheet1.Activate
19    EmpIDCard_Hide
20 End Sub
21
22 Sub GoTo_Events(control As IRibbonControl)
23    Sheet5.Activate
24 End Sub
25
26 Sub GoTo_Login(control As IRibbonControl)
27    StartWorkbook
28 End Sub
29 Sub GoTo_LogOut(control As IRibbonControl)
30    LogOut
31 End Sub
32
33 Sub GoTo_Payroll(control As IRibbonControl)
34    Sheet16.Activate
35 End Sub
36
37 Sub GoTo_Reports(control As IRibbonControl)
38    Sheet15.Activate
39 End Sub
40 Sub GoTo_ScheduleMonth(control As IRibbonControl)
41    Sheet8.Activate
42    Sched_MonthView
43 End Sub
44
45 Sub GoTo_ScheduleWeek(control As IRibbonControl)
46    Sheet8.Activate
47    Sched_WeekView
48 End Sub
49 Sub GoTo_TimeClock(control As IRibbonControl)
50    Sheet12.Activate
51 End Sub
52 Sub GoTo_TimeClockHistory(control As IRibbonControl)
53    Sheet9.Activate
54 End Sub
```

## A

Activate, [169](#)

## C

control, [169](#)

## E

EmpIDCard\_Hide, [169](#)

EmpIDCard\_Show, [169](#)

Explicit, [169](#)

## G

GoTo\_Admin, [169](#)

GoTo\_EmpArchive, [169](#)

GoTo\_EmpAttach, [169](#)

GoTo\_EmployeeID, [169](#)

GoTo\_Employees, [169](#)

GoTo\_Events, [169](#)

GoTo\_Login, [169](#)

GoTo\_LogOut, [169](#)

GoTo\_Payroll, [169](#)

GoTo\_Reports, [169](#)

GoTo\_ScheduleMonth, [169](#)

GoTo\_ScheduleWeek, [169](#)

GoTo\_TimeClock, [169](#)

GoTo\_TimeClockHistory, [169](#)

## I

IRibbonControl, [169](#)

## L

LogOut, [169](#)

## S

Sched\_MonthView, [169](#)

Sched\_WeekView, [169](#)

Sheet1, [169](#)

Sheet12, [169](#)

Sheet15, [169](#)

Sheet16, [169](#)

Sheet3, [169](#)

Sheet4, [169](#)

Sheet5, [169](#)

Sheet6, [169](#)

Sheet8, [169](#)

Sheet9, [169](#)

StartWorkbook, [169](#)



```
1 Option Explicit
2 Dim SelLeaveRow As Long
3 Sub CreateLeaveEvent()
4     Dim SelRow As Long
5     Dim SelCol As Long
6     Dim SelColumns As Long
7     Dim DateRow As Long
8     Dim EmpRow As Long
9     SelRow = ActiveCell.Row
10    SelCol = ActiveCell.Column
11    SelColumns = Selection.Columns.Count
12    If SelCol < 40 Then DateRow = 4 Else: DateRow = 5
13    EmpRow = ActiveCell.Row - 4 'Set Employee Row
14    With Sheet5
15        .Activate
16        Event_AddNew 'Set Events To Add New
17        .Range("E4").Value = Sheet2.Range("C" & EmpRow).Value & " " & Sheet4.Range("G" &
18        SelLeaveRow).Value & " Leave" 'Build Event Leave Name
19        .Range("H4").Value = "Leave"
20        .Range("K4").Value = Sheet8.Range("D" & SelRow).Value 'Full Name
21        .Range("J10").Value = Sheet8.Cells(DateRow, SelCol).Value 'Start Date
22        .Range("J11").Value = Sheet8.Cells(DateRow, SelCol + SelColumns - 1).Value 'Set
23        Ending On Date
24        .Range("N10").Value = Sheet4.Range("G" & SelLeaveRow).Value 'Leave Type
25        .Range("E6").Select
26    End With
27 End Sub
28 Sub CreateLeaveEvent10()
29     SelLeaveRow = 10
30     CreateLeaveEvent
31 End Sub
32 Sub CreateLeaveEvent11()
33     SelLeaveRow = 11
34     CreateLeaveEvent
35 End Sub
36 Sub CreateLeaveEvent12()
37     SelLeaveRow = 12
38     CreateLeaveEvent
39 End Sub
40 Sub CreateLeaveEvent13()
41     SelLeaveRow = 13
42     CreateLeaveEvent
43 End Sub
44 Sub CreateLeaveEvent14()
45     SelLeaveRow = 14
46     CreateLeaveEvent
47 End Sub
48 Sub CreateLeaveEvent15()
49     SelLeaveRow = 15
50     CreateLeaveEvent
51 End Sub
52 Sub CreateLeaveEvent16()
53     SelLeaveRow = 16
54     CreateLeaveEvent
55 End Sub
56 Sub CreateLeaveEvent17()
57     SelLeaveRow = 17
58     CreateLeaveEvent
59 End Sub
60 Sub CreateLeaveEvent18()
61     SelLeaveRow = 18
```

1

```

1
60 CreateLeaveEvent
61 End Sub
62 Sub CreateLeaveEvent19()
63     SelLeaveRow = 19
64     CreateLeaveEvent
65 End Sub
66 Sub CreateLeaveEvent20()
67     SelLeaveRow = 20
68     CreateLeaveEvent
69 End Sub
70 Sub CreateLeaveEvent21()
71     SelLeaveRow = 21
72     CreateLeaveEvent
73 End Sub
74 Sub CreateLeaveEvent22()
75     SelLeaveRow = 22
76     CreateLeaveEvent
77 End Sub
78 Sub CreateLeaveEvent23()
79     SelLeaveRow = 23
80     CreateLeaveEvent
81 End Sub
82
83
84 Sub CreateLeaveEvent7()
85     SelLeaveRow = 7
86     CreateLeaveEvent
87 End Sub
88 Sub CreateLeaveEvent8()
89     SelLeaveRow = 8
90     CreateLeaveEvent
91 End Sub
92 Sub CreateLeaveEvent9()
93     SelLeaveRow = 9
94     CreateLeaveEvent
95 End Sub
96
97 Sub RightClick_CallEvents()
98     Dim LeaveTypeRow As Long
99     Dim LeaveBtn As CommandBarPopup
100     Dim GoToEventBtn As CommandBarButton
101     Dim EditTimeClockBtn As CommandBarButton
102     With Sheet4
103         On Error Resume Next
104         With Application
105             .CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
106             .CommandBars("Cell" ).Controls("Go To Event" ).Delete
107             .CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
108             Set LeaveBtn = .CommandBars("Cell" ).Controls.Add(Before:=1, Type:=msoControlPopup)
109             Set GoToEventBtn = .CommandBars("Cell" ).Controls.Add(Before:=2, Type:=
110             msoControlButton)
111             Set EditTimeClockBtn = .CommandBars("Cell" ).Controls.Add(Before:=3, Type:=
112             msoControlButton)
113         End With
114         With LeaveBtn
115             .Caption = "Add Leave Event"
116             For LeaveTypeRow = 7 To 23
117                 If Sheet4.Range("G" & LeaveTypeRow).Value = "" Then Exit For
118                 With .Controls.Add(Type:=msoControlButton)
119                     .Caption = Sheet4.Range("G" & LeaveTypeRow).Value
120                     .FaceId = 612

```

1 2 3 4 5

```
1 2 3 4 5
119     .OnAction = "CreateLeaveEvent" & LeaveTypeRow
120     End With
121     Next LeaveTypeRow
122     End With
123     With GoToEventBtn
124         .Caption = "Go To Event"
125         .FaceId = 1812
126         .OnAction = "Sched_GoToEvent"
127     End With
128     With EditTimeClockBtn
129         .Caption = "Edit Time Clock Entry"
130         .FaceId = 33
131         .OnAction = "TimeClock_EditEntry"
132     End With
133     End With
134 End Sub
135 Sub RightClick_Cancel()
136     On Error Resume Next
137     With Application
138         .CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
139         .CommandBars("Cell" ).Controls("Go To Event" ).Delete
140         .CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
141     End With
142     On Error GoTo 0
143 End Sub
```

**A**

Activate, [171](#)  
ActiveCell, [171](#)  
Add, [172](#)  
Application, [172](#), [173](#)

**B**

Before, [172](#)

**C**

Caption, [172](#), [173](#)  
Cells, [171](#)  
Column, [171](#)  
Columns, [171](#)  
CommandBarButton, [172](#)  
CommandBarPopup, [172](#)  
CommandBars, [172](#), [173](#)  
Controls, [172](#), [173](#)  
Count, [171](#)  
CreateLeaveEvent, [171](#), [172](#)  
CreateLeaveEvent10, [171](#)  
CreateLeaveEvent11, [171](#)  
CreateLeaveEvent12, [171](#)  
CreateLeaveEvent13, [171](#)  
CreateLeaveEvent14, [171](#)  
CreateLeaveEvent15, [171](#)  
CreateLeaveEvent16, [171](#)  
CreateLeaveEvent17, [171](#)  
CreateLeaveEvent18, [171](#)  
CreateLeaveEvent19, [172](#)  
CreateLeaveEvent20, [172](#)  
CreateLeaveEvent21, [172](#)  
CreateLeaveEvent22, [172](#)  
CreateLeaveEvent23, [172](#)  
CreateLeaveEvent7, [172](#)  
CreateLeaveEvent8, [172](#)  
CreateLeaveEvent9, [172](#)

**D**

DateRow, [171](#)  
Delete, [172](#), [173](#)

**E**

EditTimeClockBtn, [172](#), [173](#)  
EmpRow, [171](#)  
Event\_AddNew, [171](#)  
Explicit, [171](#)

**F**

FacelId, [172](#), [173](#)

**G**

GoToEventBtn, [172](#), [173](#)

**L**

LeaveBtn, [172](#)  
LeaveTypeRow, [172](#), [173](#)

**M**

msoControlButton, [172](#)  
msoControlPopup, [172](#)

**O**

OnAction, [173](#)

**R**

Range, [171](#), [172](#)  
RightClick\_CallEvents, [172](#)  
RightClick\_Cancel, [173](#)

Row, [171](#)

**S**

SelCol, [171](#)  
SelColumns, [171](#)  
Selection, [171](#)  
SelLeaveRow, [171](#), [172](#)  
SelRow, [171](#)  
Sheet2, [171](#)  
Sheet4, [171](#), [172](#)  
Sheet5, [171](#)  
Sheet8, [171](#)

**V**

Value, [171](#), [172](#)

## Option Explicit

```

1  Option Explicit
2
3  Sub Sched_AddComment_Event()
4      Dim ActRow As Long
5      Dim ActCol As Long
6      Dim EventID As Long
7      Dim EventRow As Long
8      Dim EventIDRng, FoundEventIDRng As Range
9      Dim EventName, EventType, EventEmpl As String
10     Dim EventNotes, EventRecur, EventRecQty, EventRecFreq As String
11     Dim EventCreatedOn, StartOn, EndOn As Date
12     Dim EventComment As String
13     With Sheet8
14         .Range("F8:DG314").ClearComments 'Clear All Comments
15         ActRow = ActiveCell.Row
16         ActCol = ActiveCell.Column
17         If .Cells(ActRow, ActCol + 60).Value = Empty Then Exit Sub
18         EventID = .Cells(ActRow, ActCol + 60).Value
19         Set EventIDRng = Sheet7.Range("EventID")
20         Set FoundEventIDRng = EventIDRng.Find(EventID, , xlValues, xlWhole)
21         If Not FoundEventIDRng Is Nothing Then
22             EventRow = FoundEventIDRng.Row
23             EventName = Sheet7.Range("B" & EventRow).Value 'Event Name
24             EventType = Sheet7.Range("C" & EventRow).Value 'Event Type
25             EventEmpl = Sheet7.Range("D" & EventRow).Value 'Event Employee
26             EventNotes = Sheet7.Range("G" & EventRow).Value 'Event Notes
27             EventRecur = Sheet7.Range("H" & EventRow).Value 'Recurring Option
28             EventRecQty = Sheet7.Range("I" & EventRow).Value 'Recurring Qty
29             EventRecFreq = Sheet7.Range("J" & EventRow).Value 'Recurring Frequency
30             EventCreatedOn = Sheet7.Range("F" & EventRow).Value 'Event Created On Date & Time
31             StartOn = Sheet7.Range("K" & EventRow).Value + Sheet7.Range("L" & EventRow).Value
32             EndOn = Sheet7.Range("M" & EventRow).Value + Sheet7.Range("N" & EventRow).Value
33             EventComment = "Event Name: " & EventName & vbCrLf & "Created On: " &
34             EventCreatedOn & vbCrLf &
35             & "Type: " & EventType & vbCrLf & "Employee: " & EventEmpl & vbCrLf & "Event
36             Notes: " & EventNotes & vbCrLf
37             'If Recurring Add in Details
38             If EventRecur = "Yes" Then EventComment = EventComment & "Rekurs Every: " &
39             EventRecQty & " " & EventRecFreq & vbCrLf 'Add On Recurring
40             EventComment = EventComment & "Start On: " & StartOn & vbCrLf & "End On: " & EndOn
41
42             ActiveCell.ClearComments
43             With ActiveCell.AddComment
44                 With ActiveCell.Comment
45                     .Shape.Width = 240
46                     .Shape.Height = 200
47                     .Shape.AutoShapeType = msoShapeRoundedRectangle
48                     .Shape.TextFrame.Characters.Font.Name = "Tahoma"
49                     .Shape.TextFrame.Characters.Font.Size = 11
50                     .Shape.TextFrame.Characters.Font.Bold = False
51                     .Shape.TextFrame.Characters.Font.ColorIndex = 2
52                     .Shape.Line.ForeColor.RGB = RGB(0, 0, 0)
53                     .Shape.Line.BackColor.RGB = RGB(255, 255, 255)
54                     .Shape.Fill.Visible = msoTrue
55                     .Shape.Fill.ForeColor.RGB = RGB(58, 82, 184)
56                     .Shape.Fill.OneColorGradient msoGradientDiagonalUp, 1, 0.23
57                     .Text Text:=EventComment
58                 End With
59             End With
60         End With
61     End With
62 End Sub

```

```

56   1 2 3 End If
57   ActiveCell.Comment.Visible = True
58   End With
59 End Sub
60
61 Sub Sched_AddComment_Holiday()
62     Dim HolidayDt As String
63     Dim HolidayRow As Long
64     Dim HolidayName As String
65     Dim HoldDtRng As Range
66     Dim FoundHoldDtRng As Range
67     Set HoldDtRng = Sheet4.Range("Holiday_Dates" )
68     HolidayDt = Sheet8.Cells(4, ActiveCell.Column).Value 'Holiday Date
69     Set FoundHoldDtRng = HoldDtRng.Find(HolidayDt, , xlValues, xlWhole)
70     If Not FoundHoldDtRng Is Nothing Then
71         Sheet8.Range("F8:AK314" ).ClearComments 'Clear All Comments
72         HolidayRow = FoundHoldDtRng.Row
73
74         HolidayName = Sheet4.Range("D" & HolidayRow).Value
75         ActiveCell.ClearComments
76         With ActiveCell.AddComment
77             With ActiveCell.Comment
78                 .Shape.Width = 200
79                 .Shape.Height = 20
80                 .Shape.AutoShapeType = msoShapeRoundedRectangle
81                 .Shape.TextFrame.Characters.Font.Name = "Tahoma"
82                 .Shape.TextFrame.Characters.Font.Size = 11
83                 .Shape.TextFrame.Characters.Font.Bold = False
84                 .Shape.TextFrame.Characters.Font.ColorIndex = 2
85                 .Shape.Line.ForeColor.RGB = RGB(0, 0, 0)
86                 .Shape.Line.BackColor.RGB = RGB(255, 255, 255)
87                 .Shape.Fill.Visible = msoTrue
88                 .Shape.Fill.ForeColor.RGB = RGB(102, 0, 204)
89                 .Shape.Fill.OneColorGradient msoGradientDiagonalUp, 1, 0.23
90                 .Text Text:="Holiday: " & HolidayName
91             End With
92         End With
93     End If
94 End Sub
95 Sub Sched_ClearFilters()
96     With Sheet8
97         .Range("DD3,T3" ).Value = "[All Types/Positions]"
98         .Range("DF3,AC3" ).Value = "[All Event Types]"
99         If ActiveSheet.AutoFilterMode Then
100            .AutoFilter.ShowAllData
101            .AutoFilterMode = False
102        End If
103    End With
104 End Sub
105 Sub Sched_GoToEvent()
106     With Sheet8
107         Dim ActRow As Long
108         Dim ActCol As Long
109         Dim EventID As Long
110         ActRow = ActiveCell.Row
111         ActCol = ActiveCell.Column
112         If .Cells(ActRow, ActCol + 60).Value = Empty Then
113             MsgBox "Please select a valid event to open"
114             Exit Sub
115         End If
116         EventID = .Cells(ActRow, ActCol + 60).Value 'Event ID

```

1 2

```
1 2
117 Sheet5.Activate
118 Sheet5.Range("B4" ).Value = EventID
119 Dim EventRow As Long
120 Dim EventRng As Range
121 Set EventRng = Sheet5.Range("027:0999" ).Find(EventID, , xlValues, xlWhole)
122 If Not EventRng Is Nothing Then Sheet5.Range("B10" ).Value = EventRng.Row
123 Event_Load 'Run Load Macro Event
124 End With
125 End Sub
126
127 Sub Sched_MonthView()
128 With Sheet8
129 StopCalc
130 .Shapes("MonthGrp" ).Visible = msoCTrue
131 .Shapes("WeekGrp" ).Visible = msoFalse
132 .Range("F:CZ" ).EntireColumn.Hidden = False
133 .Range("DA:F0" ).EntireColumn.Hidden = True
134 ActiveWindow.FreezePanes = False
135 Range("F8" ).Select
136 ActiveWindow.FreezePanes = True
137
138 ResetCalc
139 End With
140 End Sub
141
142 Sub Sched_NextMonth()
143 Sched_ClearFilters
144 With Sheet8
145 .Range("F8:AK314" ).ClearComments 'Clear All Comments
146 If .Range("B1" ).Value = 12 Then
147 .Range("E2" ).Value = .Range("E2" ).Value + 1
148 .Range("B1" ).Value = 1
149 Else: .Range("B1" ).Value = .Range("B1" ).Value + 1
150 End If
151 End With
152 Sched_Refresh
153 End Sub
154
155 Sub Sched_NextWeek()
156 Sched_ClearFilters
157 With Sheet8
158 .Range("B2" ).Value = .Range("B2" ).Value + 7
159 If Year(.Range("B2" ).Value) <> .Range("E2" ).Value Then .Range("E2" ).Value = .Range(
" E2" ).Value + 1
160 .Range("F8:DG314" ).ClearComments 'Clear All Comments
161 Sched_Refresh
162 End With
163 End Sub
164
165 Sub Sched_PrevMonth()
166 Sched_ClearFilters
167 With Sheet8
168 .Range("F8:AK314" ).ClearComments 'Clear All Comments
169 If .Range("B1" ).Value = 1 Then
170 .Range("E2" ).Value = .Range("E2" ).Value - 1
171 .Range("B1" ).Value = 12
172 Else: .Range("B1" ).Value = .Range("B1" ).Value - 1
173 End If
174 End With
175 Sched_Refresh
176 End Sub
```

```
177
178 Sub Sched_PrevWeek()
179     With Sheet8
180         .Range("B2" ).Value = .Range("B2" ).Value - 7
181         If Year(.Range("B2" ).Value) <> .Range("E2" ).Value Then .Range("E2" ).Value = .Range(
182             "E2" ).Value - 1
183         .Range("F8:DG314" ).ClearComments 'Clear All Comments
184     End With
185 End Sub
186 Sub Sched_Refresh()
187     With Sheet8
188         StopCalc
189         .Range("F6:F06" ).Copy
190         .Range("F8:F0314" ).PasteSpecial xlPasteFormulas
191         .Calculate
192         .Range("F8:F0314" ).Copy
193         .Range("F8:F0314" ).PasteSpecial xlPasteValues
194         Application.CutCopyMode = False
195         .Range("E2" ).Select
196         .Range("F8:DG314" ).ClearComments 'Clear All Comments
197     End With
198 End Sub
199 Sub Sched_ScrollToRow()
200     ActiveWindow.ScrollRow = Sheet8.Range("B3" ).Value
201 End Sub
202 Sub Sched_ShowHideCapacity()
203     If Sheet8.Range("B7" ).Value = True Then Sheet8.Range("7:7" ).EntireRow.Hidden = False
204     Else: Sheet8.Range("7:7" ).EntireRow.Hidden = True
205 End Sub
206
207 Sub Sched_ThisMonth()
208     Sched_ClearFilters
209     Sheet8.Range("E2" ).Value = Year(Now)
210     Sheet8.Range("B1" ).Value = Month(Now)
211     Sched_Refresh
212 End Sub
213
214 Sub Sched_ThisWeek()
215     Sched_ClearFilters
216     Sheet8.Range("B2" ).Value = Sheet8.Range("A11" ).Value
217     Sheet8.Range("E2" ).Value = Year(Date)
218     Sched_Refresh
219 End Sub
220 Sub Sched_WeekView()
221     With Sheet8
222         StopCalc
223         .Shapes("MonthGrp" ).Visible = msoFalse
224         .Shapes("WeekGrp" ).Visible = msoCTrue
225         .Range("F:CZ" ).EntireColumn.Hidden = True
226         .Range("DA:F0" ).EntireColumn.Hidden = False
227         ActiveWindow.FreezePanes = False
228         Range("DA8" ).Select
229         ActiveWindow.FreezePanes = True
230         .Range("F8:DG309" ).ClearComments 'Clear All Comments
231     End With
232 End Sub
233
```



→, 175

## A

ActCol, 175, 176  
 Activate, 177  
 ActiveCell, 175, 176  
 ActiveSheet, 176  
 ActiveWindow, 177, 178  
 ActRow, 175, 176  
 AddComment, 175, 176  
 Application, 178  
 AutoFilter, 176  
 AutoFilterMode, 176  
 AutoShapeType, 175, 176

## B

BackColor, 175, 176  
 Bold, 175, 176

## C

Calculate, 178  
 Cells, 175, 176  
 Characters, 175, 176  
 ClearComments, 175-178  
 ColorIndex, 175, 176  
 Column, 175, 176  
 Comment, 175, 176  
 Copy, 178  
 CutCopyMode, 178

## E

Empty, 175, 176  
 EndOn, 175  
 EntireColumn, 177, 178  
 EntireRow, 178  
 Event\_Load, 177  
 EventComment, 175  
 EventCreatedOn, 175  
 EventEmpl, 175  
 EventID, 175-177  
 EventIDRng, 175  
 EventName, 175  
 EventNotes, 175  
 EventRecFreq, 175  
 EventRecQty, 175  
 EventRecur, 175  
 EventRng, 177  
 EventRow, 175, 177  
 EventType, 175  
 Explicit, 175

## F

Fill, 175, 176  
 Find, 175-177  
 Font, 175, 176  
 ForeColor, 175, 176  
 FoundEventIDRng, 175  
 FoundHolDtRng, 176  
 FreezePanels, 177, 178

## H

Height, 175, 176  
 Hidden, 177, 178  
 HolDtRng, 176  
 HolidayDt, 176  
 HolidayName, 176  
 HolidayRow, 176

## L

Line, 175, 176

## M

Month, 178  
 MsgBox, 176  
 msoCTrue, 177, 178  
 msoFalse, 177, 178  
 msoGradientDiagonalUp, 175, 176  
 msoShapeRoundedRectangle, 175, 176  
 msoTrue, 175, 176

## N

Name, 175, 176  
 Now, 178

## O

OneColorGradient, 175, 176

## P

PasteSpecial, 178

## R

Range, 175-178  
 ResetCalc, 177, 178  
 RGB, 175, 176  
 Row, 175-177

## S

Sched\_AddComment\_Event, 175  
 Sched\_AddComment\_Holiday, 176  
 Sched\_ClearFilters, 176-178  
 Sched\_GoToEvent, 176  
 Sched\_MonthView, 177  
 Sched\_NextMonth, 177  
 Sched\_NextWeek, 177  
 Sched\_PrevMonth, 177  
 Sched\_PrevWeek, 178  
 Sched\_Refresh, 177, 178  
 Sched\_ScrollToRow, 178  
 Sched\_ShowHideCapacity, 178  
 Sched\_ThisMonth, 178  
 Sched\_ThisWeek, 178  
 Sched\_WeekView, 178  
 ScrollRow, 178  
 Shape, 175, 176  
 Shapes, 177, 178  
 Sheet4, 176  
 Sheet5, 177  
 Sheet7, 175  
 Sheet8, 175-178  
 ShowAllData, 176  
 Size, 175, 176  
 StartOn, 175  
 StopCalc, 177, 178

## T

Text, 175, 176  
 TextFrame, 175, 176

## V

Value, 175-178  
 vbCrLf, 175  
 Visible, 175-178

## W

Width, 175, 176

## X

xlPasteFormulas, 178  
 xlPasteValues, 178  
 xlValues, 175-177  
 xlWhole, 175-177

## Y

Year, 177, 178

```
1 Option Explicit
2 Sub AMPMSel()
3     If ActiveSheet.Shapes("AMPM").TextFrame.Characters.Text = "AM" Then PMSel Else: AMSel
4 End Sub
5 Sub AMSel()
6     UnProtTP
7     With ActiveSheet
8         .Shapes("AMPM").TextFrame.Characters.Text = "AM"
9         .Shapes("AMAct").Visible = True
10        .Shapes("AMInact").Visible = False
11        .Shapes("PMAct").Visible = False
12        .Shapes("PMinact").Visible = True
13    End With
14    ProtTP
15 End Sub
16
17 Sub DoneBtn()
18     With ActiveSheet
19         ActiveCell.Value = .Shapes("HrsDisp").TextFrame.Characters.Text & ":" & .Shapes(
20             "MinDisp").TextFrame.Characters.Text & " " & .Shapes("AMPM").TextFrame.Characters.
21             Text
22         .Shapes("TimePickGrp").Visible = False
23         'SetHideShow
24     End With
25 End Sub
26
27 Sub GrpHrs()
28     Dim GrpRange, GrpShapes
29     On Error GoTo ExitSub
30     Set GrpShapes = ActiveSheet.Shapes.Range(Array("1Hr", "2Hr", "3Hr", "4Hr", "5Hr",
31         "6Hr", "7Hr", "8Hr", "9Hr", "10Hr", "11Hr", "12Hr" ))
32     Set GrpRange = GrpShapes.Group
33     GrpRange.Name = "HrsGrp"
34     ExitSub:
35 End Sub
36
37 Sub GrpMin()
38     Dim GrpRange, GrpShapes
39     On Error GoTo ExitSub
40     Set GrpShapes = ActiveSheet.Shapes.Range(Array("00Min", "05Min", "10Min", "15Min",
41         "20Min", "25Min", "30Min", "35Min", "40Min", "45Min", "50Min", "55Min" ))
42     Set GrpRange = GrpShapes.Group
43     GrpRange.Name = "MinGrp"
44     ExitSub:
45 End Sub
46
47 Sub GrpSet()
48     Dim GrpRange, GrpShapes
49     On Error GoTo ExitSub
50     Set GrpShapes = ActiveSheet.Shapes.Range(Array("5Pie1", "5Pie2" ))
51     Set GrpRange = GrpShapes.Group
52     GrpRange.Name = "5Thm"
53     Set GrpShapes = ActiveSheet.Shapes.Range(Array("4Pie1", "4Pie2" ))
54     Set GrpRange = GrpShapes.Group
55     GrpRange.Name = "4Thm"
56     Set GrpShapes = ActiveSheet.Shapes.Range(Array("3Pie1", "3Pie2" ))
57     Set GrpRange = GrpShapes.Group
58     GrpRange.Name = "3Thm"
59     Set GrpShapes = ActiveSheet.Shapes.Range(Array("2Pie1", "2Pie2" ))
60     Set GrpRange = GrpShapes.Group
61     GrpRange.Name = "2Thm"
62     Set GrpShapes = ActiveSheet.Shapes.Range(Array("1Pie1", "1Pie2" ))
63     Set GrpRange = GrpShapes.Group
64     GrpRange.Name = "1Thm"
```

1

```
1
58 Set GrpShapes = ActiveSheet.Shapes.Range(Array("5MinPie" , "5MinTxt" ))
59 Set GrpRange = GrpShapes.Group
60 GrpRange.Name = "5MinGrp"
61 Set GrpShapes = ActiveSheet.Shapes.Range(Array("10MinPie" , "10MinTxt" ))
62 Set GrpRange = GrpShapes.Group
63 GrpRange.Name = "10MinGrp"
64 Set GrpShapes = ActiveSheet.Shapes.Range(Array("15MinPie" , "15MinTxt" ))
65 Set GrpRange = GrpShapes.Group
66 GrpRange.Name = "15MinGrp"
67 Set GrpShapes = ActiveSheet.Shapes.Range(Array("SetBack" , "1Thm" , "2Thm" , "3Thm" ,
68 "4Thm" , "5Thm" , "IntTxt" , "ThmTxt" , "5MinGrp" , "10MinGrp" , "15MinGrp" ))
69 Set GrpRange = GrpShapes.Group
70 GrpRange.Name = "SetGrp"
71 ExitSub:
72 End Sub
73 Sub HideHrSel()
74 With ActiveSheet
75 .Shapes("1Sel" ).Visible = False
76 .Shapes("2Sel" ).Visible = False
77 .Shapes("3Sel" ).Visible = False
78 .Shapes("4Sel" ).Visible = False
79 .Shapes("5Sel" ).Visible = False
80 .Shapes("6Sel" ).Visible = False
81 .Shapes("7Sel" ).Visible = False
82 .Shapes("8Sel" ).Visible = False
83 .Shapes("9Sel" ).Visible = False
84 .Shapes("10Sel" ).Visible = False
85 .Shapes("11Sel" ).Visible = False
86 .Shapes("12Sel" ).Visible = False
87 End With
88 End Sub
89 Sub HideMinSel()
90 With ActiveSheet
91 .Shapes("05MSel" ).Visible = False
92 .Shapes("10MSel" ).Visible = False
93 .Shapes("15MSel" ).Visible = False
94 .Shapes("20MSel" ).Visible = False
95 .Shapes("25MSel" ).Visible = False
96 .Shapes("30MSel" ).Visible = False
97 .Shapes("35MSel" ).Visible = False
98 .Shapes("40MSel" ).Visible = False
99 .Shapes("45MSel" ).Visible = False
100 .Shapes("50MSel" ).Visible = False
101 .Shapes("55MSel" ).Visible = False
102 .Shapes("00MSel" ).Visible = False
103 End With
104 End Sub
105 Sub HoursDisplay()
106 Dim Siz, CurLeft, CurTop As Long
107 With ActiveSheet
108 UnProtTP
109 HideMinSel
110 HideHrSel
111 .Shapes("MinDisp" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255)
112 .Shapes("HrsDisp" ).TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 0, 0)
113 If ActiveCell.Value = Empty Then 'Set Default Time to 12PM
114 .Shapes("HrsDisp" ).TextFrame.Characters.Text = "12"
115 .Shapes("MinDisp" ).TextFrame.Characters.Text = "00"
116 .Shapes("AMPM" ).TextFrame.Characters.Text = "PM"
117 End If
118 .Shapes("MinGrp" ).Visible = False
```

1 2

```

1 2
118 With .Shapes("HrsGrp" )
119     .Visible = True
120     CurLeft = .Left
121     CurTop = .Top
122     For Siz = 180 To 152 Step -4
123         .Width = Siz
124         .Height = Siz
125         .Left = 152 - Siz + CurLeft
126         .Top = 152 - Siz + CurTop
127         Application.Wait (Now + 0#)
128     Next Siz
129 End With
130 On Error Resume Next
131 .Shapes(.Shapes("HrsDisp" ).TextFrame.Characters.Text & "Sel" ).Visible = True
132 On Error GoTo 0
133 ProtTP
134 End With
135 End Sub
136 Sub MacroLinkRemover()
137     'PURPOSE: Remove an external workbook reference from all shapes triggering macros
138     'Source: www.ExcelForFreelancers.com
139     Dim shp As Shape
140     Dim MacroLink, NewLink As String
141     Dim SplitLink As Variant
142
143     For Each shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
144
145         'Grab current macro link (if available)
146         On Error GoTo NextShp
147         MacroLink = shp.OnAction
148
149         'Determine if shape was linking to a macro
150         If MacroLink <> "" And InStr(MacroLink, "!") <> 0 Then
151             'Split Macro Link at the exclamation mark (store in Array)
152             SplitLink = Split(MacroLink, "!")
153
154             'Pull text occurring after exclamation mark
155             NewLink = SplitLink(1)
156
157             'Remove any stragglng apostrophes from workbook name
158             If Right(NewLink, 1) = "'" Then
159                 NewLink = Left(NewLink, Len(NewLink) - 1)
160             End If
161
162             'Apply New Link
163             shp.OnAction = NewLink
164         End If
165         NextShp:
166     Next shp
167 End Sub
168
169
170
171
172
173 Sub MinDisplay()
174     Dim Siz, CurLeft, CurTop As Long
175     Dim Inter As String
176     Dim Intg5, Intg10, Intg15 As Integer
177     With ActiveSheet
178         UnProtTP

```

```

1 2
179 Intg5 = .Shapes("5MinTxt").Fill.Visible
180 Intg10 = .Shapes("10MinTxt").Fill.Visible
181 Intg15 = .Shapes("15MinTxt").Fill.Visible
182 If Intg5 = -1 Then Inter = "5M"
183 If Intg10 = -1 Then Inter = "10M"
184 If Intg15 = -1 Then Inter = "15M"
185
186 HideHrSel
187 .Shapes("HrsDisp").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 255, 255)
188 .Shapes("MInDisp").TextFrame2.TextRange.Font.Fill.ForeColor.RGB = RGB(255, 0, 0)
189 .Shapes("HrsGrp").Visible = False
190 .Shapes("MinGrp").Visible = True
191 If Inter = "10M" Or Inter = "15M" Then .Shapes("05Min").Visible = False
192 If Inter = "15M" Then .Shapes("10Min").Visible = False
193 If Inter = "10M" Then .Shapes("15Min").Visible = False
194 If Inter = "15M" Then .Shapes("20Min").Visible = False
195 If Inter = "10M" Or Inter = "15M" Then .Shapes("25Min").Visible = False
196 If Inter = "10M" Or Inter = "15M" Then .Shapes("35Min").Visible = False
197 If Inter = "15M" Then .Shapes("40Min").Visible = False
198 If Inter = "10M" Then .Shapes("45Min").Visible = False
199 If Inter = "15M" Then .Shapes("50Min").Visible = False
200 If Inter = "10M" Or Inter = "15M" Then .Shapes("55Min").Visible = False
201 With .Shapes("MinGrp")
202     .Visible = True
203     CurLeft = .Left
204     CurTop = .Top
205     For Siz = 126 To 152 Step 2
206         .Width = Siz
207         .Height = Siz
208         .Left = 152 - Siz + CurLeft
209         .Top = 152 - Siz + CurTop
210         Application.Wait (Now + 0#)
211     Next Siz
212 End With
213 On Error Resume Next
214 .Shapes(.Shapes("MinDisp").TextFrame.Characters.Text & "MSel").Visible = True
215 On Error GoTo 0
216 ProtTP
217 End With
218 End Sub
219 Sub PMSel()
220     UnProtTP
221     With ActiveSheet
222         .Shapes("AMPM").TextFrame.Characters.Text = "PM"
223         .Shapes("AMAct").Visible = False
224         .Shapes("AMInact").Visible = True
225         .Shapes("PMAct").Visible = True
226         .Shapes("PMInact").Visible = False
227     End With
228     ProtTP
229 End Sub
230 Sub ProtTP()
231     Dim GrpRange, GrpShapes
232     On Error GoTo ExitSub
233     Set GrpShapes = ActiveSheet.Shapes.Range(Array("MainBack", "BackCircle", "HrsGrp",
234         "8Sel", _
235         "9Sel", "10Sel", "11Sel", "12Sel", "1Sel", "2Sel", "3Sel", "4Sel", "5Sel",
236         "HrsDisp", "Colon", "MinDisp", "AMPM", "AMAct", "PMInact", "DivLine",
237         "DoneBtn", _
238         "SettingsBtn", "6Sel", "7Sel", "MinGrp", "PMAct", "AMInact", "00MSel", _

```

1

```
1
237     "05Sel" , "10Sel" , "15Sel" , "20Sel" , "25Sel" , "30Sel" , "35Sel" ,
    "40Sel" , _
238     "45Sel" , "50Sel" , "55Sel" , "SetGrp" ))
239 Set GrpRange = GrpShapes.Group
240 GrpRange.Name = "TimePickGrp"
241 ActiveSheet.Shapes("TimePickGrp").Placement = 2
242 ExitSub:
243 'Add in any Protect sheet code here such as ActiveSheet.Protect "Password"
244 'Application.ScreenUpdating = True
245 End Sub
246
247 Sub Select1()
248     UnProtTP
249     HideHrSel
250     With ActiveSheet
251         .Shapes("1Sel").Visible = True
252         .Shapes("HrsDisp").TextFrame.Characters.Text = "1"
253         MinDisplay
254     End With
255     ProtTP
256 End Sub
257 Sub Select10()
258     UnProtTP
259     HideHrSel
260     With ActiveSheet
261         .Shapes("10Sel").Visible = True
262         .Shapes("HrsDisp").TextFrame.Characters.Text = "10"
263         MinDisplay
264     End With
265     ProtTP
266 End Sub
267 Sub Select11()
268     UnProtTP
269     HideHrSel
270     With ActiveSheet
271         .Shapes("11Sel").Visible = True
272         .Shapes("HrsDisp").TextFrame.Characters.Text = "11"
273         MinDisplay
274     End With
275     ProtTP
276 End Sub
277 Sub Select12()
278     UnProtTP
279     HideHrSel
280     With ActiveSheet
281         .Shapes("12Sel").Visible = True
282         .Shapes("HrsDisp").TextFrame.Characters.Text = "12"
283         MinDisplay
284     End With
285     ProtTP
286 End Sub
287 Sub Select2()
288     UnProtTP
289     HideHrSel
290     With ActiveSheet
291         .Shapes("2Sel").Visible = True
292         .Shapes("HrsDisp").TextFrame.Characters.Text = "2"
293         If .Shapes("HrsGrp").Visible = True Then MinDisplay
294     End With
295     ProtTP
296 End Sub
```

```
297 Sub Select3()  
298     UnProtTP  
299     HideHrSel  
300     With ActiveSheet  
301         .Shapes("3Sel" ).Visible = True  
302         .Shapes("HrsDisp" ).TextFrame.Characters.Text = "3"  
303         MinDisplay  
304     End With  
305     ProtTP  
306 End Sub  
307 Sub Select4()  
308     UnProtTP  
309     HideHrSel  
310     With ActiveSheet  
311         .Shapes("4Sel" ).Visible = True  
312         .Shapes("HrsDisp" ).TextFrame.Characters.Text = "4"  
313         MinDisplay  
314     End With  
315     ProtTP  
316 End Sub  
317 Sub Select5()  
318     UnProtTP  
319     HideHrSel  
320     With ActiveSheet  
321         .Shapes("5Sel" ).Visible = True  
322         .Shapes("HrsDisp" ).TextFrame.Characters.Text = "5"  
323         MinDisplay  
324     End With  
325     ProtTP  
326 End Sub  
327 Sub Select6()  
328     UnProtTP  
329     HideHrSel  
330     With ActiveSheet  
331         .Shapes("6Sel" ).Visible = True  
332         .Shapes("HrsDisp" ).TextFrame.Characters.Text = "6"  
333         MinDisplay  
334     End With  
335     ProtTP  
336 End Sub  
337 Sub Select7()  
338     UnProtTP  
339     HideHrSel  
340     With ActiveSheet  
341         .Shapes("7Sel" ).Visible = True  
342         .Shapes("HrsDisp" ).TextFrame.Characters.Text = "7"  
343         MinDisplay  
344     End With  
345     ProtTP  
346 End Sub  
347 Sub Select8()  
348     UnProtTP  
349     HideHrSel  
350     With ActiveSheet  
351         .Shapes("8Sel" ).Visible = True  
352         .Shapes("HrsDisp" ).TextFrame.Characters.Text = "8"  
353         MinDisplay  
354     End With  
355     ProtTP  
356 End Sub  
357 Sub Select9()
```

1

```
1
358 UnProtTP
359 HideHrSel
360 With ActiveSheet
361     .Shapes("9Sel" ).Visible = True
362     .Shapes("HrsDisp" ).TextFrame.Characters.Text = "9"
363     MinDisplay
364 End With
365 ProtTP
366 End Sub
367 Sub SelectM00()
368     UnProtTP
369     HideMinSel
370     With ActiveSheet
371         .Shapes("00MSel" ).Visible = True
372         .Shapes("MinDisp" ).TextFrame.Characters.Text = "00"
373     End With
374     ProtTP
375 End Sub
376 Sub SelectM05()
377     UnProtTP
378     HideMinSel
379     With ActiveSheet
380         .Shapes("05MSel" ).Visible = True
381         .Shapes("MinDisp" ).TextFrame.Characters.Text = "05"
382     End With
383     ProtTP
384 End Sub
385 Sub SelectM10()
386     UnProtTP
387     HideMinSel
388     With ActiveSheet
389         .Shapes("10MSel" ).Visible = True
390         .Shapes("MinDisp" ).TextFrame.Characters.Text = "10"
391     End With
392     ProtTP
393 End Sub
394 Sub SelectM15()
395     UnProtTP
396     HideMinSel
397     With ActiveSheet
398         .Shapes("15MSel" ).Visible = True
399         .Shapes("MinDisp" ).TextFrame.Characters.Text = "15"
400     End With
401     ProtTP
402 End Sub
403 Sub SelectM20()
404     UnProtTP
405     HideMinSel
406     With ActiveSheet
407         .Shapes("20MSel" ).Visible = True
408         .Shapes("MinDisp" ).TextFrame.Characters.Text = "20"
409     End With
410     ProtTP
411 End Sub
412 Sub SelectM25()
413     UnProtTP
414     HideMinSel
415     With ActiveSheet
416         .Shapes("25MSel" ).Visible = True
417         .Shapes("MinDisp" ).TextFrame.Characters.Text = "25"
418     End With
```

1



```
1
419     ProtTP
420 End Sub
421 Sub SelectM30()
422     UnProtTP
423     HideMinSel
424     With ActiveSheet
425         .Shapes("30MSel" ).Visible = True
426         .Shapes("MinDisp" ).TextFrame.Characters.Text = "30"
427     End With
428     ProtTP
429 End Sub
430 Sub SelectM35()
431     UnProtTP
432     HideMinSel
433     With ActiveSheet
434         .Shapes("35MSel" ).Visible = True
435         .Shapes("MinDisp" ).TextFrame.Characters.Text = "35"
436     End With
437     ProtTP
438 End Sub
439 Sub SelectM40()
440     UnProtTP
441     HideMinSel
442     With ActiveSheet
443         .Shapes("40MSel" ).Visible = True
444         .Shapes("MinDisp" ).TextFrame.Characters.Text = "40"
445     End With
446     ProtTP
447 End Sub
448 Sub SelectM45()
449     UnProtTP
450     HideMinSel
451     With ActiveSheet
452         .Shapes("45MSel" ).Visible = True
453         .Shapes("MinDisp" ).TextFrame.Characters.Text = "45"
454     End With
455     ProtTP
456 End Sub
457 Sub SelectM50()
458     UnProtTP
459     HideMinSel
460     With ActiveSheet
461         .Shapes("50MSel" ).Visible = True
462         .Shapes("MinDisp" ).TextFrame.Characters.Text = "50"
463     End With
464     ProtTP
465 End Sub
466 Sub SelectM55()
467     UnProtTP
468     HideMinSel
469     With ActiveSheet
470         .Shapes("55MSel" ).Visible = True
471         .Shapes("MinDisp" ).TextFrame.Characters.Text = "55"
472     End With
473     ProtTP
474 End Sub
475
476 Sub Set10MinInt()
477     With ActiveSheet
478         .Shapes("10MinTxt" ).Fill.ForeColor.RGB = RGB(140, 58, 58)
479         .Shapes("10MinTxt" ).Fill.Visible = msoCTrue
1 2
```

```
1 2
480     .Shapes("5MinTxt" ).Fill.Visible = msoFalse
481     .Shapes("15MinTxt" ).Fill.Visible = msoFalse
482 End With
483     MinDisplay
484     SelectM00
485 End Sub
486
487 Sub Set15MinInt()
488     With ActiveSheet
489         .Shapes("15MinTxt" ).Fill.ForeColor.RGB = RGB(140, 58, 58)
490         .Shapes("15MinTxt" ).Fill.Visible = msoCTrue
491         .Shapes("10MinTxt" ).Fill.Visible = msoFalse
492         .Shapes("5MinTxt" ).Fill.Visible = msoFalse
493     End With
494     MinDisplay
495     SelectM00
496 End Sub
497 Sub Set5MinInt()
498     With ActiveSheet
499         .Shapes("5MinTxt" ).Fill.ForeColor.RGB = RGB(140, 58, 58)
500         .Shapes("5MinTxt" ).Fill.Visible = msoCTrue
501         .Shapes("10MinTxt" ).Fill.Visible = msoFalse
502         .Shapes("15MinTxt" ).Fill.Visible = msoFalse
503     End With
504     MinDisplay
505     SelectM00
506 End Sub
507
508 Sub SetHideShow()
509     With ActiveSheet
510         UnProtTP
511         If .Shapes("SetGrp" ).Visible = False Then
512             .Shapes("SetGrp" ).Visible = msoCTrue
513         Else
514             .Shapes("SetGrp" ).Visible = False
515         End If
516     End With
517
518     ProtTP
519 End Sub
520
521 Sub SetThm1()
522     ActiveSheet.Shapes.Range(Array("SetBack" , "MainBack" )).Fill.ForeColor.RGB = RGB(64, 64,
523         64)
524     ActiveSheet.Shapes("BackCircle" ).Fill.ForeColor.RGB = RGB(54, 54, 54)
525 End Sub
526 Sub SetThm2()
527     ActiveSheet.Shapes.Range(Array("SetBack" , "MainBack" )).Fill.ForeColor.RGB = RGB(55, 96,
528         146)
529     ActiveSheet.Shapes("BackCircle" ).Fill.ForeColor.RGB = RGB(37, 64, 97)
530 End Sub
531 Sub SetThm3()
532     ActiveSheet.Shapes.Range(Array("SetBack" , "MainBack" )).Fill.ForeColor.RGB = RGB(119, 147
533         , 60)
534     ActiveSheet.Shapes("BackCircle" ).Fill.ForeColor.RGB = RGB(79, 98, 40)
535 End Sub
536 Sub SetThm4()
537     ActiveSheet.Shapes.Range(Array("SetBack" , "MainBack" )).Fill.ForeColor.RGB = RGB(96, 74,
538         123)
539     ActiveSheet.Shapes("BackCircle" ).Fill.ForeColor.RGB = RGB(64, 49, 82)
540 End Sub
```

```

537 Sub SetThm5()
538     ActiveSheet.Shapes.Range(Array("SetBack" , "MainBack" )).Fill.ForeColor.RGB = RGB(49, 133,
156)
539     ActiveSheet.Shapes("BackCircle" ).Fill.ForeColor.RGB = RGB(33, 89, 104)
540 End Sub
541 Sub TPShow()
542     Dim SelCell As Range
543     Dim Hr As Long
544     'On Error GoTo NoTP
545     'If Sheet is Protected, add Unprotect Code here such as ActiveSheet.Unprotect "Password"
546     With ActiveSheet.Shapes("TimePickGrp" )
547         Set SelCell = Selection
548         .Visible = msoCTrue
549         ' On Error Resume Next
550         .Left = SelCell.Left
551         .Top = SelCell.Offset(1, 0).Top
552         ' On Error GoTo 0
553     End With
554     UnProtTP
555     HideHrSel
556     HideMinSel
557     'Check for Links from another workbook
558     UnGrpSet
559     If InStr(ActiveSheet.Shapes("5Pie1" ).OnAction, "!") <> 0 Then 'Run Workbook Link
560         Remover
561         UnGrpHrs
562         UnGrpMin
563         MacroLinkRemover
564         GrpHrs
565         GrpMin
566         GrpSet
567     Else:
568         GrpSet
569     End If
570     If ActiveCell.Value <> Empty Then Hr = Format(ActiveCell.Value, "hh" )
571     If Hr >= 13 Then Hr = Hr - 12
572     If Hr = 0 Then Hr = 12
573     If ActiveCell.Value <> Empty Then ActiveSheet.Shapes("HrsDisp" ).TextFrame.Characters.Text
574     = Hr
575     If ActiveCell.Value <> Empty Then ActiveSheet.Shapes("MInDisp" ).TextFrame.Characters.Text
576     = Right(Format(ActiveCell.Value, "h:mm" ), 2)
577     If ActiveCell.Value <> Empty Then ActiveSheet.Shapes("AMPM" ).TextFrame.Characters.Text =
578     Format(ActiveCell.Value, "AM/PM" )
579     ActiveSheet.Shapes("SetGrp" ).Visible = msoFalse
580     HoursDisplay
581     ProtTP
582     'If Sheet is Protected, Add Protection Code here such as ActiveSheet.Protect "Password"
583     Exit Sub
584     NoTP:
585     MsgBox "The Time Picker has been removed from this sheet. Please copy the entire Time
586     Picker from another sheet and paste it into this sheet"
587     'If Sheet is Protected, Add Protection Code here such as ActiveSheet.Protect "Password"
588 End Sub
589
590 Sub UnGrpHrs()
591     On Error Resume Next
592     ActiveSheet.Shapes("HrsGrp" ).Ungroup
593     On Error GoTo 0
594 End Sub

```

```
592 Sub UnGrpMin()  
593     On Error Resume Next  
594     ActiveSheet.Shapes("MinGrp").Ungroup  
595     On Error GoTo 0  
596 End Sub  
597 Sub UnGrpSet()  
598     On Error Resume Next  
599     ActiveSheet.Shapes("SetGrp").Ungroup  
600     ActiveSheet.Shapes("5Thm").Ungroup  
601     ActiveSheet.Shapes("4Thm").Ungroup  
602     ActiveSheet.Shapes("3Thm").Ungroup  
603     ActiveSheet.Shapes("2Thm").Ungroup  
604     ActiveSheet.Shapes("1Thm").Ungroup  
605     ActiveSheet.Shapes("5MinGrp").Ungroup  
606     ActiveSheet.Shapes("10MinGrp").Ungroup  
607     ActiveSheet.Shapes("15MinGrp").Ungroup  
608     On Error GoTo 0  
609 End Sub  
610 Sub UnProtTP()  
611     On Error Resume Next  
612     ActiveSheet.Shapes("TimePickGrp").Ungroup  
613     On Error GoTo 0  
614     'Add in any Unprotect sheet code here such as ActiveSheet.Unprotect "Password"  
615     'Application.ScreenUpdating = False  
616 End Sub
```

\_, 183, 184

## A

ActiveCell, 180, 181, 189  
 ActiveSheet, 180-190  
 AMPMSel, 180  
 AMSel, 180  
 Application, 182, 183  
 Array, 180, 181, 183, 188, 189

## C

Characters, 180-187, 189  
 CurLeft, 181-183  
 CurTop, 181-183

## D

DoneBtn, 180

## E

Empty, 181, 189  
 ExitSub, 180, 181, 183, 184  
 Explicit, 180

## F

Fill, 181, 183, 187-189  
 Font, 181, 183  
 ForeColor, 181, 183, 187-189  
 Format, 189

## G

Group, 180, 181, 184  
 GrpHrs, 180, 189  
 GrpMin, 180, 189  
 GrpRange, 180, 181, 183, 184  
 GrpSet, 180, 189  
 GrpShapes, 180, 181, 183, 184

## H

Height, 182, 183  
 HideHrSel, 181, 183-186, 189  
 HideMinSel, 181, 186, 187, 189  
 HoursDisplay, 181, 189  
 Hr, 189

## I

InStr, 182, 189  
 Inter, 182, 183  
 Intg10, 182, 183  
 Intg15, 182, 183  
 Intg5, 182, 183

## L

Left, 182, 183, 189  
 Len, 182

## M

MacroLink, 182  
 MacroLinkRemover, 182, 189  
 MinDisplay, 182, 184-186, 188  
 MsgBox, 189  
 msoCTrue, 187-189  
 msoFalse, 188, 189

## N

Name, 180, 181, 184  
 NewLink, 182  
 NextShp, 182  
 NoTP, 189

Now, 182, 183

## O

Offset, 189  
 OnAction, 182, 189

## P

Placement, 184  
 PMSel, 180, 183  
 ProtTP, 180, 182-189

## R

Range, 180, 181, 183, 188, 189  
 RGB, 181, 183, 187-189  
 Right, 182, 189

## S

SelCell, 189  
 Select1, 184  
 Select10, 184  
 Select11, 184  
 Select12, 184  
 Select2, 184  
 Select3, 185  
 Select4, 185  
 Select5, 185  
 Select6, 185  
 Select7, 185  
 Select8, 185  
 Select9, 185  
 Selection, 189  
 SelectM00, 186, 188  
 SelectM05, 186  
 SelectM10, 186  
 SelectM15, 186  
 SelectM20, 186  
 SelectM25, 186  
 SelectM30, 187  
 SelectM35, 187  
 SelectM40, 187  
 SelectM45, 187  
 SelectM50, 187  
 SelectM55, 187  
 Set10MinInt, 187  
 Set15MinInt, 188  
 Set5MinInt, 188  
 SetHideShow, 188  
 SetThm1, 188  
 SetThm2, 188  
 SetThm3, 188  
 SetThm4, 188  
 SetThm5, 189  
 Shape, 182  
 Shapes, 180-190  
 shp, 182  
 Siz, 181-183  
 Split, 182  
 SplitLink, 182

## T

Text, 180-187, 189  
 TextFrame, 180-187, 189  
 TextFrame2, 181, 183  
 TextRange, 181, 183  
 Top, 182, 183, 189  
 TPShow, 189

## U

Ungroup, 189, 190  
 UnGrpHrs, 189

UnGrpMin, 189, 190  
 UnGrpSet, 189, 190  
 UnProtTP, 180-190

## V

Value, 180, 181, 189  
 Visible, 180-189

## W

Wait, 182, 183  
 Width, 182, 183

```

1 Option Explicit
2 Dim TCRow As Long
3
4 Sub EditTimeClock_Calculate()
5     Dim ClockInVal As Double
6     Dim ClockOutVal As Double
7     Dim BreakOutVal As Double
8     Dim BreakInVal As Double
9
10    With EditTimeClock
11        If .ClockIn.Value <> "" Then ClockInVal = Format(.ClockIn.Value, "0.00" )
12        If .BreakOut.Value <> "" Then BreakOutVal = Format(.BreakOut.Value, "0.00" )
13        If .BreakIn.Value <> "" Then BreakInVal = Format(.BreakIn.Value, "0.00" )
14        If .ClockOut.Value <> "" Then ClockOutVal = Format(.ClockOut.Value, "0.00" )
15
16        If ClockInVal = Empty Or ClockOutVal = Empty Then
17            .RegHrsValue.Caption = 0
18            .OTHrsValue.Caption = 0
19        Else:
20            If BreakInVal = 0 Or BreakOutVal = 0 Then 'If either of the breaks are missing,
                then set both with zero value
21                BreakInVal = 0
22                BreakOutVal = 0
23            End If
24            If ((ClockOutVal - ClockInVal) - (BreakInVal - BreakOutVal)) * 24 > Sheet4.Range(
                "MaxDayHrs" ).Value Then 'Overtime
25                .RegHrsValue.Caption = Sheet4.Range("MaxDayHrs" ).Value 'Set Max Day Hours
26                .OTHrsValue.Caption = ((ClockOutVal - ClockInVal) - (BreakInVal - BreakOutVal)) *
                24 - Sheet4.Range("MaxDayHrs" ).Value
27            Else: 'No Overtime
28                .RegHrsValue.Caption = ((ClockOutVal - ClockInVal) - (BreakInVal - BreakOutVal))
                * 24
29                .OTHrsValue.Caption = 0
30            End If
31
32            .RegHrsValue.Caption = Format(.RegHrsValue.Caption, "0.0" ) 'Set Total Regular
                Hours
33            .OTHrsValue.Caption = Format(.OTHrsValue.Caption, "0.0" ) 'Set Total OT Hours
34        End If
35    End With
36
37 End Sub
38
39
40 Sub EditTimeClock_Save()
41     Dim SelRow As Long
42     With EditTimeClock
43         If .ClockIn.Value = Empty Then
44             MsgBox "Please make sure at least a Clock In exists for this Time Clock Entry"
45             Exit Sub
46         End If
47
48         If ActiveSheet.CodeName = "Sheet16" Then 'Payroll Timeclock Edit
49             SelRow = Sheet16.Range("B14" ).Value
50             TCRow = Sheet16.Range("AK" & SelRow).Value 'Timeclock Row in Payroll Screen
51         Else:
52             If TCRow = 0 Then TCRow = Sheet9.Range("A99999" ).End(xlUp).Row + 1
53         End If
54         Sheet9.Range("A" & TCRow).Value = Sheet8.Range("A59" ).Value 'Time Clock ID
55         Sheet9.Range("B" & TCRow).Value = Sheet8.Range("A53" ).Value 'Employee ID
56         Sheet9.Range("C" & TCRow).Value = Sheet8.Range("A55" ).Value 'Employee Name

```

```

1 2
57 Sheet9.Range("D" & TCRow).Value = Sheet8.Range("A57" ).Value + Format(.ClockIn.Value,
    "0.00000" )
58
59
60 If .BreakOut.Value <> Empty Then Sheet9.Range("E" & TCRow).Value = Int(Sheet8.Range(
    "A57" ).Value) + Format(.BreakOut.Value, "0.00000" )
61 If .BreakOut.Value <> Empty And Sheet9.Range("E" & TCRow).Value < Sheet9.Range("D" &
    TCRow).Value Then Sheet9.Range("E" & TCRow).Value = Sheet9.Range("E" & TCRow).Value +
    1 ' Add 1 in case of overnight
62 If .BreakIn.Value <> Empty Then Sheet9.Range("F" & TCRow).Value = Sheet8.Range("A57" )
    .Value + Format(.BreakIn.Value, "0.00000" )
63 If .BreakIn.Value <> Empty And Sheet9.Range("F" & TCRow).Value < Sheet9.Range("E" &
    TCRow).Value Then Sheet9.Range("F" & TCRow).Value = Sheet9.Range("F" & TCRow).Value +
    1 ' Add 1 in case of overnight
64 If .ClockOut.Value <> Empty Then Sheet9.Range("G" & TCRow).Value = Sheet8.Range("A57"
    ).Value + Format(.ClockOut.Value, "0.00000" )
65 If .ClockOut.Value <> Empty And Sheet9.Range("G" & TCRow).Value < Sheet9.Range("F" &
    TCRow).Value Then Sheet9.Range("G" & TCRow).Value = Sheet9.Range("G" & TCRow).Value +
    1 ' Add 1 in case of overnight
66 Sheet9.Range("H" & TCRow).Value = .Notes.Value
67 Sheet9.Range("I" & TCRow).Value = "=Row()"
68 Sheet9.Range("J" & TCRow).Value = Sheet9.Range("J" & TCRow).Value & "Admin Edit: " &
    Sheet14.Range("AppUserName" ).Value & " on: " & Now 'Add Logged In Username
69 .Hide
70 End With
71 If ActiveSheet.CodeName = "Sheet16" Then 'Payroll Timeclock Edit
72     Payroll_LoadEmpReview 'Refresh Employee Schedule On Payroll Screen
73 Else:
74     Sched_Refresh 'Refresh Schedule
75 End If
76 End Sub
77
78 Sub TimeClock_EditEntry()
79     Dim SelRow As Long
80     Dim SelCol As Long
81     Dim DateRow As Long
82     Dim ClockInVal As Double
83     Dim ClockOutVal As Double
84     Dim BreakOutVal As Double
85     Dim BreakInVal As Double
86
87     SelRow = ActiveCell.Row
88     SelCol = ActiveCell.Column
89     If SelCol < 37 Then DateRow = 4 Else: DateRow = 5 'Set Date Row to 4 or 5 depending upon
    Month or Week view
90     With Sheet8
91         .Range("A53" ).Value = .Range("B" & SelRow).Value 'Add Employee ID
92         .Range("A57" ).Value = .Cells(DateRow, SelCol).Value 'Add Selected Date
93         If .Range("A51" ).Value <> Empty Then 'Existing Time Clock Row
94
95             TCRow = .Range("A51" ).Value
96             ClockInVal = Sheet9.Range("D" & TCRow).Value 'Clock In
97             BreakOutVal = Sheet9.Range("E" & TCRow).Value 'Break out
98             BreakInVal = Sheet9.Range("F" & TCRow).Value 'Break In
99             ClockOutVal = Sheet9.Range("G" & TCRow).Value 'Clock In
100         With EditTimeClock
101             If ClockInVal = 0 Then .ClockIn.Value = "" Else: .ClockIn.Value = ClockInVal
    'Clock In
102             If BreakOutVal = 0 Then .BreakOut.Value = "" Else: .BreakOut.Value = BreakOutVal
    'Clock Out Break
103             If BreakInVal = 0 Then .BreakIn.Value = "" Else: .BreakIn.Value = BreakInVal

```

1 2 3 4

```
1 2 3 4
104 'Clock In Break
    If ClockOutVal = 0 Then .ClockOut.Value = "" Else: .ClockOut.Value =
    ClockOutVal 'Clock Out
105     .Notes.Value = Sheet9.Range("H" & TCRow).Value 'Notes
106     End With
107     EditTimeClock_Calculate
108
109     Else ' New Time Clock Entry
110     With EditTimeClock 'Clear all form fields
111         .ClockIn.Value = vbNullString
112         .BreakOut.Value = vbNullString
113         .BreakIn.Value = vbNullString
114         .ClockOut.Value = vbNullString
115         .Notes.Value = vbNullString
116         .RegHrsValue.Caption = 0
117         .OTHrsValue.Caption = 0
118     End With
119     End If
120     EditTimeClock.Show 'Display Edit Time Clock form
121
122 End With
123
124 End Sub
```



**A**

ActiveCell, [193](#)  
ActiveSheet, [192](#), [193](#)

**B**

BreakIn, [192-194](#)  
BreakInVal, [192](#), [193](#)  
BreakOut, [192-194](#)  
BreakOutVal, [192](#), [193](#)

**C**

Caption, [192](#), [194](#)  
Cells, [193](#)  
ClockIn, [192-194](#)  
ClockInVal, [192](#), [193](#)  
ClockOut, [192-194](#)  
ClockOutVal, [192-194](#)  
CodeName, [192](#), [193](#)  
Column, [193](#)

**D**

DateRow, [193](#)

**E**

EditTimeClock, [192-194](#)  
EditTimeClock\_Calculate, [192](#), [194](#)  
EditTimeClock\_Save, [192](#)  
Empty, [192](#), [193](#)  
Explicit, [192](#)

**F**

Format, [192](#), [193](#)

**H**

Hide, [193](#)

**I**

Int, [193](#)

**M**

MsgBox, [192](#)

**N**

Notes, [193](#), [194](#)  
Now, [193](#)

**O**

OTHrsValue, [192](#), [194](#)

**P**

Payroll\_LoadEmpReview, [193](#)

**R**

Range, [192-194](#)  
RegHrsValue, [192](#), [194](#)  
Row, [192](#), [193](#)

**S**

Sched\_Refresh, [193](#)  
SelCol, [193](#)  
SelRow, [192](#), [193](#)  
Sheet14, [193](#)  
Sheet16, [192](#)  
Sheet4, [192](#)  
Sheet8, [192](#), [193](#)  
Sheet9, [192-194](#)  
Show, [194](#)

**T**

TCRow, [192-194](#)  
TimeClock\_EditEntry, [193](#)

**V**

Value, [192-194](#)  
vbNullString, [194](#)

**X**

xLUp, [192](#)

```
1
2 Option Explicit
3 Dim RunClk As Boolean
4 Dim EmpRow As Long
5 Sub Timeclock_AlreadyEntered()
6     Dim i As Integer
7     Dim delay As Double
8     Dim StartTime As Double
9     With ActiveSheet.Shapes("AlreadyEntered" )
10        .Visible = True
11        For i = 1 To 350
12            .Fill.Transparency = i / 350
13            delay = 0.00003
14            StartTime = Timer
15            Do
16                DoEvents
17                Loop While Timer - StartTime < delay
18            Next i
19        .Visible = False
20    End With
21 End Sub
22 Sub Timeclock_DisplayPicture()
23     Dim PicPath As String
24     With Sheet12
25         On Error Resume Next
26         .Shapes("EmplPic" ).Delete 'Delete Picture if it exists
27         On Error GoTo 0
28         PicPath = Sheet2.Range("AC" & EmpRow).Value 'Path of the picture
29         On Error GoTo NoPic
30         With .Pictures.Insert(PicPath)
31             With .ShapeRange
32                 .LockAspectRatio = msoTrue
33                 .Height = 125
34                 .Name = "EmplPic"
35             End With 'Shape Range
36         End With 'Pictures
37
38         With .Shapes("EmplPic" )
39             .Left = Sheet12.Range("K15" ).Left
40             .Top = Sheet12.Range("K15" ).Top
41             .IncrementLeft 20
42             .IncrementTop 10
43             .SoftEdge.Type = msoSoftEdgeType4
44         End With
45         NoPic:
46     End With 'Sheet 1
47
48 End Sub
49
50
51 Sub Timeclock_EnterID()
52     Dim TimeClockRow As Long
53     Dim MessRow As Long
54     Dim TimeRow As Long
55     Dim EmpID As String
56     Dim EmpName As String
57     Dim TimeClockID As String
58     Dim TimeClockID_Prev As String
59     Dim EntryType As String
60     Dim TimeClockIDRng As Range
61     Dim FoundTimeClockIDRng As Range
```

1

```

1
62 Dim FoundPrevTimeClockIDRng As Range
63 Dim DayThreshold As Double
64 Dim BreakThreshold As Double
65 Dim TotalHours As Double
66 Dim TotalBreakHours As Double
67
68 With Sheet12
69     'Hide Any Remaining Pop-Up Messages
70     .Shapes("IDNotFound").Visible = msoFalse
71     .Shapes("AlreadyEntered").Visible = msoFalse
72     .Shapes("TimeclockMessage").Visible = msoFalse
73
74
75     .Range("J13").Select 'Select Employee ID Cell to get ready for next entry
76
77     'Check For Proper Employee ID
78     If .Range("B24").Value = Empty Then
79         .Range("J13").ClearContents
80         Timeclock_IDNotFound
81         Exit Sub
82     End If
83     DayThreshold = Sheet4.Range("DayThreshold").Value
84     BreakThreshold = Sheet4.Range("BreakThreshold").Value
85
86     'Determine Clock In or Clock Out
87
88     EmpID = .Range("J13").Value
89     EmpName = .Range("B25").Value 'Employee Name
90     EmpRow = .Range("B24").Value 'Employee Row
91 End With
92 With Sheet9
93     'Build Timeclock ID's
94     TimeClockID = Year(Now) & "-" & Month(Now) & "-" & Application.WorksheetFunction.Text
95     (Day(Now), "DD") & "-" & EmpID 'Set Time Clock ID for Current Day
96     TimeClockID_Prev = Year(Now) & "-" & Month(Now) & "-" & Day(Now - 1) & "-" & EmpID
97     'Set Time Clock ID for Previous Day
98     Set TimeClockIDRng = .Range("TimeClockList_ID")
99     Set FoundTimeClockIDRng = TimeClockIDRng.Find(TimeClockID, , xlValues, xlWhole)
100
101     If FoundTimeClockIDRng Is Nothing Then 'ID Not found
102         'Before Adding a New Row, Double Check that Employee is Not working overnight shift
103         and clocked in the night before
104         Set FoundPrevTimeClockIDRng = TimeClockIDRng.Find(TimeClockID_Prev, , xlValues,
105         xlWhole)
106         If Not FoundPrevTimeClockIDRng Is Nothing Then 'Previous ID Found
107             TimeClockRow = FoundPrevTimeClockIDRng.Row 'Previous Day TimeClock Row
108
109             'Clock Out For Break On Specific 3 Conditions (Clock Out Break Must Be Empty,
110             Admin Enabled Breaks, Under Day Threshold)
111             If .Range("E" & TimeClockRow).Value = Empty And Sheet4.Range("BreakOption").
112             Value = "Yes" And (Now - .Range("D" & TimeClockRow).Value) * 24 < DayThreshold
113             Then
114                 .Range("E" & TimeClockRow).Value = Now 'Add Clock Out From Break
115                 EntryType = "BreakOut"
116                 GoTo EntryAdded
117             End If
118
119             'Clock In From Break On Specific 4 Conditions (Clock Out For Break is Not
120             Empty, Clock In From Break is Empty, Admin Enabled Breaks, and Under Break
121             Threshold )

```

1 2 3 4

```

1 2 3 4
114 If .Range("E" & TimeClockRow).Value <> Empty And .Range("F" & TimeClockRow).
Value = Empty And Sheet4.Range("BreakOption").Value = "Yes" And (Now - .Range(
"E" & TimeClockRow).Value) * 24 < BreakThreshold Then
115     .Range("F" & TimeClockRow).Value = Now 'Add Clock In From Break
116     EntryType = "BreakIn"
117     GoTo EntryAdded
118 End If
119
120 'Clock Out For the Day On Certain 2 Conditions (Clock Out is Empty and Under Day
Threshold)
121 If .Range("G" & TimeClockRow).Value = Empty And (Now - .Range("D" &
TimeClockRow).Value) * 24 < DayThreshold Then
122     .Range("G" & TimeClockRow).Value = Now
123     EntryType = "DayOut"
124     GoTo EntryAdded
125 End If
126 End If 'End Previous ID found
127
128 'ID Not Found So Add New ID and Clock In Employee
129 TimeClockRow = .Range("A999999").End(xlUp).Row + 1 'First Available Row
130 .Range("A" & TimeClockRow).Value = TimeClockID
131 .Range("B" & TimeClockRow).Value = EmpID 'Employee ID
132 .Range("C" & TimeClockRow).Value = EmpName 'Employee Name
133 .Range("D" & TimeClockRow).Value = Now 'Clock in current Time & Date
134 .Range("I" & TimeClockRow).Value = "=Row()" 'Set Row Formula
135 .Range("J" & TimeClockRow).Value = "Time Clock" 'Set Method
136 EntryType = "DayIn"
137 GoTo EntryAdded 'Skip All Other Options
138
139 Else ' TimeClock ID Found
140     TimeClockRow = FoundTimeClockIDRow.Row 'Set row of Found Time clock
141
142
143 'Clock Out For Break On Certain 3 Conditions (Clock Out Break Must Be Empty, Admin
Enabled Breaks, Under Day Threshold)
144 If .Range("E" & TimeClockRow).Value = Empty And Sheet4.Range("BreakOption").Value
= "Yes" And (Now - .Range("D" & TimeClockRow).Value) * 24 < DayThreshold Then
145     .Range("E" & TimeClockRow).Value = Now 'Add Clock Out From Break
146     EntryType = "BreakOut"
147     GoTo EntryAdded
148 End If
149
150 'Clock In From Break On Certain 4 Conditions (Clock Out For Break is Not Empty,
Clock In From Break is Empty, Admin Enabled Breaks, and Under Break Threshold )
151 If .Range("E" & TimeClockRow).Value <> Empty And .Range("F" & TimeClockRow).Value
= Empty And Sheet4.Range("BreakOption").Value = "Yes" And (Now - .Range("E" &
TimeClockRow).Value) * 24 < BreakThreshold Then
152     .Range("F" & TimeClockRow).Value = Now 'Add Clock In From Break
153     EntryType = "BreakIn"
154     GoTo EntryAdded
155 End If
156
157 'Clock Out For the Day On Certain 2 Conditions (Clock Out is Empty and Under Day
Threshold)
158 If .Range("G" & TimeClockRow).Value = Empty And (Now - .Range("D" & TimeClockRow).
Value) * 24 < DayThreshold Then
159     .Range("G" & TimeClockRow).Value = Now
160     EntryType = "DayOut"
161     GoTo EntryAdded
162 End If
163
1 2 3

```

```

1 2 3
164 End If ' End If Timeclock ID Found
165 If EntryType = Empty Then GoTo EntryAddedAlready
166 EntryAdded:
167 'Check For Completed Shift
168 If Sheet9.Range("F" & TimeClockRow).Value <> Empty Then TotalBreakHours = (Sheet9.
Range("F" & TimeClockRow).Value - Sheet9.Range("E" & TimeClockRow).Value) * 24
169 If TotalBreakHours < 0 Then TotalBreakHours = 0
170 If Sheet9.Range("G" & TimeClockRow).Value <> Empty Then TotalHours = ((Sheet9.Range(
"G" & TimeClockRow).Value - Sheet9.Range("D" & TimeClockRow).Value) * 24) -
TotalBreakHours
171
172 If EntryType = "DayIn" Then 'Set Time Row
173     If TimeValue(Now) < Sheet4.Range("E408" ).Value Then TimeRow = 1
174     If TimeValue(Now) > Sheet4.Range("E410" ).Value Then TimeRow = 3
175     If TimeRow = 0 Then TimeRow = 2
176 End If
177 If EntryType = "DayOut" Then 'Set Time Row
178     If TimeValue(Now) < Sheet4.Range("E413" ).Value Then TimeRow = 1
179     If TimeValue(Now) > Sheet4.Range("E415" ).Value Then TimeRow = 3
180     If TimeRow = 0 Then TimeRow = 2
181 End If
182
183 Select Case EntryType
184 Case Is = "BreakOut"
185     MessRow = 418
186 Case Is = "BreakIn"
187     MessRow = 419
188 Case Is = "DayIn"
189     MessRow = 407 + TimeRow
190 Case Is = "DayOut"
191     MessRow = 412 + TimeRow
192 End Select
193
194 Sheet12.Range("B26" ).Value = Replace(Replace(Replace(Replace(Sheet4.Range("F" &
MessRow).Value, "[Employee Name]" , EmpName), "[Time]" , TimeValue(Now)), "[Total
Hours]" , Round(TotalHours, 2)), "[Total Break]" , Round(TotalBreakHours, 2))
195 'Get Employee Picture if Applicable
196
197 ActiveSheet.Shapes("TimeclockMessage" ).Visible = msoCTrue
198 If Sheet4.Range("PictureOption" ).Value = "Yes" And Sheet2.Range("AC" & EmpRow).Value
<> Empty Then Timeclock_DisplayPicture
199 Application.ScreenUpdating = True
200 Application.Wait Now + 0.00007
201 ActiveSheet.Shapes("TimeclockMessage" ).Visible = msoFalse
202 On Error Resume Next
203 ActiveSheet.Shapes("EmplPic" ).Visible = msoFalse
204 On Error GoTo 0
205 Sheet12.Range("J13" ).ClearContents
206 RunClk = False
207 'Timeclock_StartStop
208 End With
209 UpdateLocal 'Update & Synch
210 Exit Sub
211 EntryAddedAlready:
212 Timeclock_AlreadyEntered
213 End Sub
214
215 Sub Timeclock_IDNotFound()
216 Dim i As Integer
217 Dim delay As Double
218 Dim StartTime As Double

```

1

```
1
219 With ActiveSheet.Shapes("IDNotFound" )
220     .Visible = True
221     For i = 1 To 300
222         .Fill.Transparency = i / 300
223         delay = 0.00004
224         StartTime = Timer
225         Do
226             DoEvents
227             Loop While Timer - StartTime < delay
228         Next i
229         .Visible = False
230     End With
231 End Sub
232 Sub Timeclock_StartStop()
233     With Sheet12
234         .Range("J13" ).Select
235         RunClk = Not (RunClk)
236         Do While RunClk = True
237             DoEvents
238             .Range("A29" ) = TimeValue(Now)
239             DoEvents
240             .Range("A30" ) = Now()
241         Loop
242     End With
243 End Sub
244
245
```

**A**

ActiveSheet, [196](#), [199](#), [200](#)  
Application, [197](#), [199](#)

**B**

BreakThreshold, [197](#), [198](#)

**C**

ClearContents, [197](#), [199](#)

**D**

Day, [197](#)  
DayThreshold, [197](#), [198](#)  
delay, [196](#), [199](#), [200](#)  
Delete, [196](#)  
DoEvents, [196](#), [200](#)

**E**

EmpID, [196-198](#)  
EmpName, [196-199](#)  
EmpRow, [196](#), [197](#), [199](#)  
Empty, [197-199](#)  
EntryAdded, [197-199](#)  
EntryAddedAlready, [199](#)  
EntryType, [196-199](#)  
Explicit, [196](#)

**F**

Fill, [196](#), [200](#)  
Find, [197](#)  
FoundPrevTimeClockIDRng, [197](#)  
FoundTimeClockIDRng, [196-198](#)

**H**

Height, [196](#)

**I**

i, [196](#), [199](#), [200](#)  
IncrementLeft, [196](#)  
IncrementTop, [196](#)  
Insert, [196](#)

**L**

Left, [196](#)  
LockAspectRatio, [196](#)

**M**

MessRow, [196](#), [199](#)  
Month, [197](#)  
msoCTrue, [199](#)  
msoFalse, [197](#), [199](#)  
msoSoftEdgeType4, [196](#)  
msoTrue, [196](#)

**N**

Name, [196](#)  
NoPic, [196](#)  
Now, [197-200](#)

**P**

PicPath, [196](#)  
Pictures, [196](#)

**R**

Range, [196-200](#)  
Replace, [199](#)  
Round, [199](#)  
Row, [197](#), [198](#)  
RunClk, [196](#), [199](#), [200](#)

**S**

ScreenUpdating, [199](#)  
ShapeRange, [196](#)  
Shapes, [196](#), [197](#), [199](#), [200](#)  
Sheet12, [196](#), [197](#), [199](#), [200](#)  
Sheet2, [196](#), [199](#)  
Sheet4, [197-199](#)  
Sheet9, [197](#), [199](#)  
SoftEdge, [196](#)  
StartTime, [196](#), [199](#), [200](#)

**T**

Text, [197](#)  
Timeclock\_AlreadyEntered, [196](#), [199](#)  
Timeclock\_DisplayPicture, [196](#), [199](#)  
Timeclock\_EnterID, [196](#)  
Timeclock\_IDNotFound, [197](#), [199](#)  
Timeclock\_StartStop, [200](#)  
TimeClockID, [196-198](#)  
TimeClockID\_Prev, [196](#), [197](#)  
TimeClockIDRng, [196](#), [197](#)  
TimeClockRow, [196-199](#)  
Timer, [196](#), [200](#)  
TimeRow, [196](#), [199](#)  
TimeValue, [199](#), [200](#)  
Top, [196](#)  
TotalBreakHours, [197](#), [199](#)  
TotalHours, [197](#), [199](#)  
Transparency, [196](#), [200](#)

**U**

UpdateLocal, [199](#)

**V**

Value, [196-199](#)  
Visible, [196](#), [197](#), [199](#), [200](#)

**W**

Wait, [199](#)  
WorksheetFunction, [197](#)

**X**

xlUp, [198](#)  
xlValues, [197](#)  
xlWhole, [197](#)

**Y**

Year, [197](#)

## Option Explicit

```

1  Option Explicit
2
3  Sub TimeClock_SyncRemoteEntries()
4      Dim fso As Object
5      Dim TimeClockRow As Long
6      Dim MainAppFolder As String
7      Dim TimeClockFolder As String
8      Dim FileName As String
9      Dim TCFile As String
10     Dim TCText As String
11     Dim TCTextLine As String
12     Dim TimeClockID As String
13     Dim EmpID As String
14     Dim ClockIn As String
15     Dim BreakOut As String
16     Dim BreakIn As String
17     Dim ClockOut As String
18     Dim TimeClockIDRng As Range
19     Dim FoundTimeClockIDRng As Range
20     Dim EmpIDRng As Range
21     Dim FoundEmpIDRng As Range
22     Set TimeClockIDRng = Sheet9.Range("TimeClockList_ID" )
23     Set EmpIDRng = Sheet2.Range("EmployeeID" )
24     With Sheet9
25         If Sheet4.Range("ApplicationFolder" ).Value = Empty Then
26             Sheet4.Activate
27             Sheet4.Range("J16" ).Select
28             MsgBox "Please make sure to select a default Dropbox (or shared) Main Application
29                 Folder in whcih timeclock entries can be stored"
30             BrowseMainApplicationFolder
31             If Sheet4.Range("ApplicationFolder" ).Value = Empty Then Exit Sub 'If there is
32                 still no folder then exit sub, otherwise continue
33         End If
34         MainAppFolder = Sheet4.Range("ApplicationFolder" ).Value 'Set Main Application Folder
35         Set fso = CreateObject("Scripting.FileSystemobject" ) 'Set File System Object
36         'Check for Time Clock Entries folder and Time Clock Archives folder and add if
37         necessary
38         If Dir(MainAppFolder & "\Time Clock Entries" , vbDirectory) = "" Then fso.createFolder
39             (MainAppFolder & "\Time Clock Entries/" )
40         If Dir(MainAppFolder & "\Time Clock Archives/" , vbDirectory) = "" Then fso.
41             createFolder (MainAppFolder & "\Time Clock Archives/" )
42         TimeClockFolder = MainAppFolder & "/Time Clock Entries/"
43         FileName = Dir(TimeClockFolder & "\*.txt" )
44
45     Do While Len(FileName) > 0 'Start of Loop
46         TCFile = TimeClockFolder & "\" & FileName
47
48         Open TCFile For Input As #1
49         Line Input #1, TCText
50         Close #1
51
52         EmpID = Mid(TCText, 13, InStr(TCText, "Clock In:" ) - 13) 'Starting Point of
53             Employee ID
54         ClockIn = Mid(TCText, InStr(TCText, "Clock In:" ) + 9, InStr(TCText, "Break Out:" )
55             - (InStr(TCText, "Clock In:" ) + 9))
56         BreakOut = Mid(TCText, InStr(TCText, "Break Out:" ) + 10, InStr(TCText, "Break In:" )
57             - (InStr(TCText, "Break Out:" ) + 10))
58         BreakIn = Mid(TCText, InStr(TCText, "Break In:" ) + 9, InStr(TCText, "Clock Out:" )
59             - (InStr(TCText, "Break In:" ) + 9))
60         ClockOut = Mid(TCText, InStr(TCText, "Clock Out:" ) + 10, (Len(TCText) - (InStr(
61             TCText, "Clock Out:" ) + 10)))

```

1 2 3



```
52     1 2 3 TimeClockID = Application.WorksheetFunction.Text(ClockIn, "yyyy-mm-dd-" ) & EmpID
53
54     Set FoundTimeClockIDRng = TimeClockIDRng.Find(TimeClockID, , xlValues, xlWhole)
55     If Not FoundTimeClockIDRng Is Nothing Then TimeClockRow = FoundTimeClockIDRng.Row
56     Else: TimeClockRow = .Range("A99999" ).End(xlUp).Row + 1 'Use Existing Found Row
57     or First Available Row
58     .Range("A" & TimeClockRow).Value = TimeClockID 'Time Clock ID
59     .Range("B" & TimeClockRow).Value = EmpID 'Employee ID
60     Set FoundEmpIDRng = EmpIDRng.Find(EmpID, , xlValues, xlWhole)
61     If Not FoundEmpIDRng Is Nothing Then .Range("C" & TimeClockRow).Value = Sheet2.
62     Range("BC" & FoundEmpIDRng.Row).Value 'Add Employee If ID is Found
63     If .Range("D" & TimeClockRow).Value = Empty Then .Range("D" & TimeClockRow).Value
64     = ClockIn 'Add Clock In Only If Existing Value is Empty
65     If .Range("E" & TimeClockRow).Value = Empty Then .Range("E" & TimeClockRow).Value
66     = BreakOut 'Add Break Out Only If Existing Value is Empty
67     If .Range("F" & TimeClockRow).Value = Empty Then .Range("F" & TimeClockRow).Value
68     = BreakIn 'Add Break Out Only If Existing Value is Empty
69     If .Range("G" & TimeClockRow).Value = Empty Then .Range("G" & TimeClockRow).Value
70     = ClockOut 'Add Break Out Only If Existing Value is Empty
71     .Range("I" & TimeClockRow).Value = "=Row()" 'Set Row #
72     .Range("J" & TimeClockRow).Value = "Remote" 'Entry Type
73
74     'Move To Time Clock Archives Folder
75
76     fso.MoveFile Source:=TCfile, Destination:=MainAppFolder & "/Time Clock Archives/" &
77     FileName 'Move To Archives Folder
78     FileName = Dir() 'Clear Curent File Name
79     Loop 'End Of Loop
80     End With
81     End Sub
```

**A**

Activate, [202](#)  
Application, [203](#)

**B**

BreakIn, [202](#), [203](#)  
BreakOut, [202](#), [203](#)  
BrowseMainApplicationFolder, [202](#)

**C**

ClockIn, [202](#), [203](#)  
ClockOut, [202](#), [203](#)  
Close, [202](#)  
createFolder, [202](#)  
CreateObject, [202](#)

**D**

Destination, [203](#)  
Dir, [202](#), [203](#)

**E**

EmpID, [202](#), [203](#)  
EmpIDRng, [202](#), [203](#)  
Empty, [202](#), [203](#)  
Explicit, [202](#)

**F**

FileName, [202](#), [203](#)  
Find, [203](#)  
FoundEmpIDRng, [202](#), [203](#)  
FoundTimeClockIDRng, [202](#), [203](#)  
fso, [202](#), [203](#)

**I**

Input, [202](#)  
InStr, [202](#)

**L**

Len, [202](#)  
Line, [202](#)

**M**

MainAppFolder, [202](#), [203](#)  
Mid, [202](#)  
MoveFile, [203](#)  
MsgBox, [202](#)

**O**

Open, [202](#)

**R**

Range, [202](#), [203](#)  
Row, [203](#)

**S**

Sheet2, [202](#), [203](#)  
Sheet4, [202](#)  
Sheet9, [202](#)  
Source, [203](#)

**T**

TCTFile, [202](#), [203](#)  
TCText, [202](#)  
TCTextLine, [202](#)  
Text, [203](#)  
TimeClock\_SyncRemoteEntries, [202](#)  
TimeClockFolder, [202](#)  
TimeClockID, [202](#), [203](#)  
TimeClockIDRng, [202](#), [203](#)

TimeClockRow, [202](#), [203](#)

**V**

Value, [202](#), [203](#)  
vbDirectory, [202](#)

**W**

WorksheetFunction, [203](#)

**X**

xLUp, [203](#)  
xlValues, [203](#)  
xlWhole, [203](#)

```
1 Option Explicit
2
3 #If VBA7 Then
4 Public Declare PtrSafe Function FindWindow Lib "user32" _
5     Alias "FindWindowA" _
6     (ByVal lpClassName As String, _
7     ByVal lpWindowName As String) As Long
8
9
10 Public Declare PtrSafe Function GetWindowLong Lib "user32" _
11     Alias "GetWindowLongA" _
12     (ByVal hWnd As Long, _
13     ByVal nIndex As Long) As Long
14
15
16 Public Declare PtrSafe Function SetWindowLong Lib "user32" _
17     Alias "SetWindowLongA" _
18     (ByVal hWnd As Long, _
19     ByVal nIndex As Long, _
20     ByVal dwNewLong As Long) As Long
21
22
23 Public Declare PtrSafe Function DrawMenuBar Lib "user32" _
24     (ByVal hWnd As Long) As Long
25 #Else
26 Public Declare Function FindWindow Lib "user32" _
27     Alias "FindWindowA" _
28     (ByVal lpClassName As String, _
29     ByVal lpWindowName As String) As Long
30
31
32 Public Declare Function GetWindowLong Lib "user32" _
33     Alias "GetWindowLongA" _
34     (ByVal hWnd As Long, _
35     ByVal nIndex As Long) As Long
36
37
38 Public Declare Function SetWindowLong Lib "user32" _
39     Alias "SetWindowLongA" _
40     (ByVal hWnd As Long, _
41     ByVal nIndex As Long, _
42     ByVal dwNewLong As Long) As Long
43
44
45 Public Declare Function DrawMenuBar Lib "user32" _
46     (ByVal hWnd As Long) As Long
47 #End If
48 Sub CheckUser()
49     Dim UserType As String
50     With Sheet14
51         .Calculate
52         If .Range("B8").Value = Empty Then 'Incorrect Username
53             MsgBox "Please enter a correct user name"
54             Exit Sub
55         End If
56
57         If .Range("B7").Value <> True Then 'Incorrect password
58             MsgBox "Pleae enter a correct password"
59             Exit Sub
60         End If
61
62     End With
63 End Sub
```

```
1 2
62     LoginForm.Hide
63     UserType = Range("B8" ).Value '.Set User Type
64     .Range("B6" ).ClearContents 'Clear Password
65     .Range("B10" ).Value = .Range("B5" ).Value 'Set Username
66     If .Range("B9" ).Value = False Then .Range("B5" ).ClearContents 'Clear Username on no
request to remember
67     If UserType = "Admin" Then User_SetAs_Admin 'Set Workbook To Admin level secutify
68     If UserType = "Manager" Then User_SetAs_Manager 'Set Workbook to manager Level
Security
69     If UserType = "User" Then User_SetAs_User 'Set Workbook to User Level security
70     If UserType = "Timeclock" Then User_SetAs_TimeClock
71     End With
72     End Sub
73
74     Sub CloseWorkbook()
75     Sheet14.Activate
76     Dim WkSht As Worksheet
77     For Each WkSht In ThisWorkbook.Worksheets
78     If WkSht.Name <> "Login" Then WkSht.Visible = xlSheetVeryHidden
79     Next WkSht
80     ThisWorkbook.Save
81     End Sub
82
83     Sub HideBar(frm As Object)
84     Dim Style As Long, Menu As Long, hWndForm As Long
85     hWndForm = FindWindow("ThunderDFrame" , frm.Caption)
86     Style = GetWindowLong(hWndForm, &HFFF0)
87     Style = Style And Not &HC0000
88     SetWindowLong hWndForm, &HFFF0, Style
89     DrawMenuBar hWndForm
90     End Sub
91
92     Sub LogOut()
93     Sheet14.Activate
94     Dim WkSht As Worksheet
95     Sheet14.Visible = xlSheetVisible
96     For Each WkSht In ThisWorkbook.Worksheets
97     If WkSht.Name <> "Login" Then WkSht.Visible = xlSheetVeryHidden
98     Next WkSht
99     'LoginForm.Show
100    End Sub
101    Sub ShowLoginForm()
102    LoginForm.Show
103    End Sub
104
105    Sub StartWorkbook()
106    Sheet14.Activate
107    LoginForm.PasswordField.Text = "" 'Clear Password
108    If Sheet14.Range("B9" ).Value = False Then LoginForm.UserNameField.Text = ""
109    LoginForm.Show
110    End Sub
```

—  
\_, 205

## A

Activate, 206

## C

Calculate, 205  
Caption, 206  
CheckUser, 205  
ClearContents, 206  
CloseWorkbook, 206

## D

DrawMenuBar, 205, 206  
dwNewLong, 205

## E

Empty, 205  
Explicit, 205

## F

FindWindow, 205, 206  
frm, 206

## G

GetWindowLong, 205, 206

## H

Hide, 206  
HideBar, 206  
hWnd, 205  
hWndForm, 206

## L

LoginForm, 206  
LogOut, 206  
lpClassName, 205  
lpWindowName, 205

## M

Menu, 206  
MsgBox, 205

## N

Name, 206  
nIndex, 205

## P

PasswordField, 206  
PtrSafe, 205

## R

Range, 205, 206

## S

Save, 206  
SetWindowLong, 205, 206  
Sheet14, 205, 206  
Sheet14Activate, 206  
Show, 206  
ShowLoginForm, 206  
StartWorkbook, 206  
Style, 206

## T

Text, 206  
ThisWorkbook, 206

## U

User\_SetAs\_Admin, 206  
User\_SetAs\_Manager, 206  
User\_SetAs\_TimeClock, 206  
User\_SetAs\_User, 206  
UserNameField, 206  
UserType, 205, 206

## V

Value, 205, 206  
VBA7, 205  
Visible, 206

## W

WkSht, 206  
Worksheet, 206  
Worksheets, 206

## X

xlSheetVeryHidden, 206  
xlSheetVisible, 206

---

1 Option Explicit

---

2  
3 Sub User\_SetAs\_Admin()

4 Sheet1.Visible = xlSheetVisible 'Show Employee Manager  
5 Sheet1.Activate  
6 Sheet2.Visible = xlSheetVisible 'Show Employee List  
7 Sheet3.Visible = xlSheetVisible 'Show Attachments List  
8 Sheet4.Visible = xlSheetVisible 'Show Admin Sheet  
9 Sheet5.Visible = xlSheetVisible 'Show Events Sheet  
10 Sheet6.Visible = xlSheetVisible 'Show Employee Archive  
11 Sheet7.Visible = xlSheetVisible 'Show Events List  
12 Sheet8.Visible = xlSheetVisible 'Show Scheduling Sheet  
13 Sheet9.Visible = xlSheetVisible 'Show Time Clock List  
14 Sheet11.Visible = xlSheetVisible 'Show Employee Leave List  
15 Sheet12.Visible = xlSheetVisible 'Show Time Clock Sheet  
16 Sheet13.Visible = xlSheetVisible 'Hide Weekly Schedule List  
17 Sheet14.Visible = xlSheetVisible ' Hide Login Sheet  
18 Sheet15.Visible = xlSheetVisible 'Show Reports  
19 Sheet16.Visible = xlSheetVisible 'Show Payroll

20 End Sub

---

21  
22  
23 Sub User\_SetAs\_Manager()

24 Sheet1.Visible = xlSheetVisible 'Show Employee Manager  
25 Sheet1.Activate  
26 Sheet2.Visible = xlSheetVisible 'Show Employee List  
27 Sheet3.Visible = xlSheetVisible 'Show Attachments List  
28 Sheet4.Visible = xlSheetVeryHidden 'Hide Admin Sheet  
29 Sheet5.Visible = xlSheetVisible 'Show Events Sheet  
30 Sheet6.Visible = xlSheetVisible 'Show Employee Archive  
31 Sheet7.Visible = xlSheetVisible 'Show Events List  
32 Sheet8.Visible = xlSheetVisible 'Show Scheduling Sheet  
33 Sheet9.Visible = xlSheetVisible 'Show Time Clock List  
34 Sheet11.Visible = xlSheetVisible 'Show Employee Leave List  
35 Sheet12.Visible = xlSheetVisible 'Show Time Clock Sheet  
36 Sheet13.Visible = xlSheetHidden 'Hide Weekly Schedule List  
37 Sheet14.Visible = xlSheetHidden ' Hide Login Sheet  
38 Sheet15.Visible = xlSheetVisible 'Show Reports  
39 Sheet16.Visible = xlSheetVisible 'Show Payroll

40 End Sub

---

41  
42 Sub User\_SetAs\_TimeClock()

43 Dim WkSht As Worksheet  
44 Sheet12.Activate 'Activate Time Clock Sheet  
45 For Each WkSht In ThisWorkbook.Worksheets  
46 If WkSht.Name <> "Time Clock" Then WkSht.Visible = xlSheetHidden  
47 Next WkSht  
48 Sheet12.Range("J13" ).Select

49 End Sub

---

50  
51  
52  
53  
54  
55 Sub User\_SetAs\_User()

56 Sheet1.Visible = xlSheetVisible 'Show Employee Manager  
57 Sheet1.Activate  
58 Sheet2.Visible = xlSheetVisible 'Show Employee List  
59 Sheet3.Visible = xlSheetVisible 'Show Attachments List  
60 Sheet4.Visible = xlSheetVeryHidden 'Hide Admin Sheet  
61 Sheet5.Visible = xlSheetVisible 'Show Events Sheet

1

```
1
62 Sheet6.Visible = xlSheetVisible 'Show Employee Archive
63 Sheet7.Visible = xlSheetVisible 'Show Events List
64 Sheet8.Visible = xlSheetVisible 'Show Scheduling Sheet
65 Sheet9.Visible = xlSheetVisible 'Show Time Clock List
66 Sheet11.Visible = xlSheetVisible 'Show Employee Leave List
67 Sheet12.Visible = xlSheetVisible 'Show Time Clock Sheet
68 Sheet13.Visible = xlSheetHidden 'Hide Weekly Schedule List
69 Sheet14.Visible = xlSheetHidden ' Hide Login Sheet
70 Sheet15.Visible = xlSheetVeryHidden 'Show Reports
71 Sheet16.Visible = xlSheetVeryHidden 'Show Payroll
72 End Sub
```

## A

Activate, [208](#)

## E

Explicit, [208](#)

## N

Name, [208](#)

## R

Range, [208](#)

## S

Sheet1, [208](#)  
Sheet11, [208](#), [209](#)  
Sheet12, [208](#), [209](#)  
Sheet13, [208](#), [209](#)  
Sheet14, [208](#), [209](#)  
Sheet15, [208](#), [209](#)  
Sheet16, [208](#), [209](#)  
Sheet2, [208](#)  
Sheet3, [208](#)  
Sheet4, [208](#)  
Sheet5, [208](#)  
Sheet6, [208](#), [209](#)  
Sheet7, [208](#), [209](#)  
Sheet8, [208](#), [209](#)  
Sheet9, [208](#), [209](#)

## T

ThisWorkbook, [208](#)

## U

User\_SetAs\_Admin, [208](#)  
User\_SetAs\_Manager, [208](#)  
User\_SetAs\_TimeClock, [208](#)  
User\_SetAs\_User, [208](#)

## V

Visible, [208](#), [209](#)

## W

WkSht, [208](#)  
Worksheet, [208](#)  
Worksheets, [208](#)

## X

xlSheetHidden, [208](#), [209](#)  
xlSheetVeryHidden, [208](#), [209](#)  
xlSheetVisible, [208](#), [209](#)



# 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*