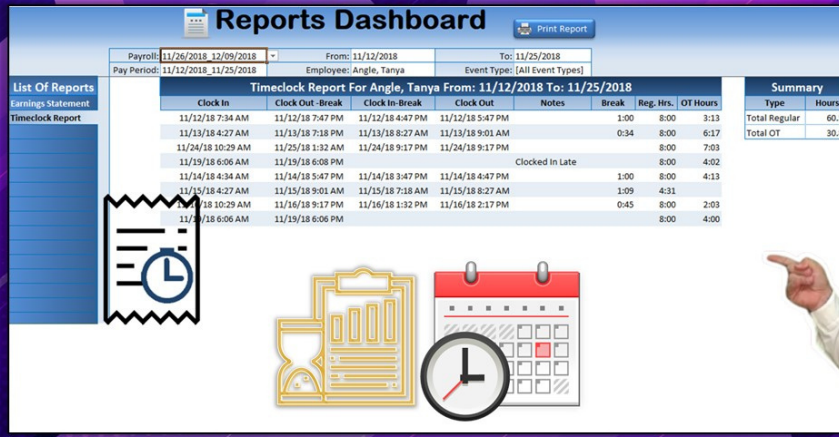


# VBA SOURCE CODE BOOK

## EMPLOYEE MANAGER PT.19 TIMECLOCK REPORT



Timeclock Report For Angle, Tanya From: 11/12/2018 To: 11/25/2018							Summary		
Clock In	Clock Out - Break	Clock In - Break	Clock Out	Notes	Break	Reg. Hrs.	OT Hours	Type	Hours
11/12/18 7:34 AM	11/12/18 7:47 PM	11/12/18 4:47 PM	11/12/18 5:47 PM		1:00	8:00	3:13	Total Regular	60.52
11/13/18 4:27 AM	11/13/18 7:18 PM	11/13/18 8:27 AM	11/13/18 9:01 AM		0:34	8:00	6:17	Total OT	30.85
11/24/18 10:29 AM	11/25/18 1:32 AM	11/24/18 9:17 PM	11/24/18 9:17 PM			8:00	7:03		
11/19/18 6:06 AM	11/19/18 6:08 PM			Clocked In Late		8:00	4:02		
11/14/18 4:34 AM	11/14/18 5:47 PM	11/14/18 3:47 PM	11/14/18 4:47 PM		1:00	8:00	4:13		
11/15/18 4:27 AM	11/15/18 9:01 AM	11/15/18 7:18 AM	11/15/18 8:27 AM		1:09	4:31			
11/16/18 10:29 AM	11/16/18 9:17 PM	11/16/18 1:32 PM	11/16/18 2:17 PM		0:45	8:00	2:03		
11/19/18 6:06 AM	11/19/18 6:06 PM					8:00	4:00		



*Excel For Freelancers*

How To Create A Dynamic  
Timeclock Report in Excel  
Today  
[Employee Manager Pt. 19]



[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



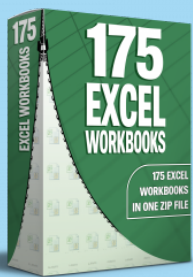
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[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.....	8
VBAProject .....	8
Documents .....	8
Sheet1.....	8
(Declarations).....	8
Worksheet_Activate [Sub ].....	8
Worksheet_Change [Sub ].....	8
Worksheet_Deactivate [Sub ].....	10
Worksheet_SelectionChange [Sub ] .....	10
Sheet10.....	13
(Declarations).....	13
Sheet11.....	15
(Declarations).....	15
Sheet12.....	17
Worksheet_Change [Sub ] .....	17
Sheet13.....	19
(Declarations).....	19
Sheet14.....	21
(Declarations).....	21
TextBox1_Change [Sub ] .....	21
Sheet15.....	23
(Declarations).....	23
Worksheet_Change [Sub ] .....	23
Worksheet_SelectionChange [Sub ] .....	23
Sheet16.....	25
(Declarations).....	25
Worksheet_Change [Sub ] .....	25
Worksheet_SelectionChange [Sub ] .....	25
Sheet17.....	28
(Declarations).....	28
Sheet18.....	30
(Declarations).....	30
Sheet19.....	32
(Declarations).....	32
Sheet2.....	34
(Declarations).....	34
Sheet20.....	36
(Declarations).....	36
Sheet3.....	38
(Declarations).....	38
Worksheet_Change [Sub ] .....	38
Sheet4.....	40
(Declarations).....	40
Worksheet_Activate [Sub ].....	40
Worksheet_Change [Sub ] .....	40
Worksheet_Deactivate [Sub ].....	40
Worksheet_SelectionChange [Sub ] .....	40
Sheet5.....	43
(Declarations).....	43
Worksheet_Activate [Sub ].....	43
Worksheet_BeforeDoubleClick [Sub ] .....	43
Worksheet_Change [Sub ] .....	43
Worksheet_Deactivate [Sub ].....	44
Worksheet_SelectionChange [Sub ] .....	44
Sheet6.....	47
(Declarations).....	47
Worksheet_SelectionChange [Sub ] .....	47
Sheet7.....	49
(Declarations).....	49
Sheet8.....	51
(Declarations).....	51
Worksheet_Activate [Sub ].....	51
Worksheet_Change [Sub ] .....	51
Worksheet_SelectionChange [Sub ] .....	51
Sheet9.....	53
(Declarations).....	53

ThisWorkbook	55
(Declarations)	55
Workbook_Activate [Sub ]	55
Workbook_BeforeClose [Sub ]	55
Workbook_Deactivate [Sub ]	55
Workbook_SheetBeforeRightClick [Sub ]	55
Forms	57
CreatePayPeriod	57
(Declarations)	57
CancelBtn_Click [Sub ]	57
ClsBtn_Click [Sub ]	57
CreateBtn_Click [Sub ]	57
EditTimeClock	59
(Declarations)	59
BreakIn_AfterUpdate [Sub ]	59
BreakInNow_MouseDown [Sub ]	59
BreakOut_AfterUpdate [Sub ]	59
BreakOutNow_MouseDown [Sub ]	59
CancelBtn_MouseDown [Sub ]	59
ClockIn_AfterUpdate [Sub ]	59
ClockInNow_MouseDown [Sub ]	59
ClockOut_AfterUpdate [Sub ]	59
ClockOutNow_MouseDown [Sub ]	59
OkBtn_Click [Sub ]	59
UserForm_Activate [Sub ]	60
LoginForm	62
(Declarations)	62
CloseBtn_Click [Sub ]	62
OkBtn_Click [Sub ]	62
UserForm_Initialize [Sub ]	62
Modules	64
Admin_EventNotif	64
(Declarations)	64
Notif_AddEmailAttach [Sub ]	64
Notif_AddNew [Sub ]	64
Notif_CancelNew [Sub ]	64
Notif_Delete [Sub ]	64
Notif_Load [Sub ]	64
Notif_Save [Sub ]	64
Admin_Macros	67
(Declarations)	67
Admin_SetTotalFormat [Sub ]	67
BrowseEmployeeFolder [Sub ]	67
BrowseRemoteTimeClockFolder [Sub ]	67
CreatePayPeriods [Sub ]	67
ShowPayPeriodForm [Sub ]	68
Admin_Tabs	70
(Declarations)	70
Admin_SwitchHorizontalTabs [Sub ]	70
AutoTab_Macros	72
do_nothing [Sub ]	72
GetColLtr [Function ]	72
GetTabOrder [Function ]	72
SetOnkey [Sub ]	72
TabRange [Sub ]	73
UpOrDownArrow [Sub ]	73
CodeReset_Macs	76
(Declarations)	76
ResetCalc [Sub ]	76
StopCalc [Sub ]	76
Empl_Attach	78
(Declarations)	78
Attach_Add [Sub ]	78
Attach_Delete [Sub ]	78
Attach_DisplayThumbnail [Sub ]	79
Attach_Filter [Sub ]	80

Attach_FilterClear [Sub ] .....	80
Attach_Open [Sub ] .....	80
Attach_Refresh [Sub ] .....	80
Attach_ShowAll [Sub ] .....	81
Empl_IDCard .....	83
(Declarations) .....	83
EmplIDCard_CreateTextBox [Sub ] .....	83
EmplIDCard_Customize [Sub ] .....	83
EmplIDCard_FieldChange [Sub ] .....	83
EmplIDCard_Hide [Sub ] .....	83
EmplIDCard_HideCustom [Sub ] .....	83
EmplIDCard_HidePic [Sub ] .....	84
EmplIDCard_Print [Sub ] .....	84
EmplIDCard_Show [Sub ] .....	84
EmplIDCard_ShowPic [Sub ] .....	84
Empl_LoadMacros .....	86
(Declarations) .....	86
Empl_EventsRefresh [Sub ] .....	86
Empl_LeaveRefresh [Sub ] .....	86
Empl_Load [Sub ] .....	86
Empl_PayItemsRefresh [Sub ] .....	87
Empl_TimeclockRefresh [Sub ] .....	88
Empl_WeeklySchedRefresh [Sub ] .....	88
Empl_Misc .....	91
(Declarations) .....	91
Empl_Archive [Sub ] .....	91
Empl_CancelNew [Sub ] .....	91
Empl_New [Sub ] .....	91
Empl_ReActivate [Sub ] .....	92
Empl_SaveNew [Sub ] .....	92
Empl_SortByName [Sub ] .....	93
Empl_PicsMacs .....	95
(Declarations) .....	95
Add_EmplPic [Sub ] .....	95
Show_EmplPic [Sub ] .....	95
Empl_Tabs .....	98
(Declarations) .....	98
Empl_SwitchHorizontalTabs [Sub ] .....	98
EmpTab_Events [Sub ] .....	98
EmpTab_GenInfo [Sub ] .....	98
EmpTab_Leave [Sub ] .....	98
EmpTab_PayDetails [Sub ] .....	98
EmpTab_PayHistory [Sub ] .....	98
EmpTab_Scheduling [Sub ] .....	98
EmpTab_TimeClock [Sub ] .....	99
SwitchVerticalTabs [Sub ] .....	99
Events_Filters .....	101
(Declarations) .....	101
Events_ClearFilter [Sub ] .....	101
Events_FilterTable [Sub ] .....	101
Events_Misc .....	103
(Declarations) .....	103
DeleteFilters [Sub ] .....	103
Event_AddEmailAttach [Sub ] .....	103
Event_AddNew [Sub ] .....	103
Event_AddNotifDefaults [Sub ] .....	103
Event_AllDay [Sub ] .....	103
Event_CancelNew [Sub ] .....	103
Event_Delete [Sub ] .....	103
Event_Load [Sub ] .....	104
Event_Refresh [Sub ] .....	104
Event_SaveNew [Sub ] .....	105
FadeOutMessage .....	107
(Declarations) .....	107
EmployIDNotFound [Sub ] .....	107
FolderPickMac .....	109

(Declarations).....	109
GetFolder [Sub ].....	109
MailToSelected [Sub ].....	109
Notification_Macros .....	111
(Declarations).....	111
Notification_Close [Sub ].....	111
Notifications_Check [Sub ].....	111
Notifications_Create [Sub ].....	111
testting [Sub ].....	113
Payroll_Load.....	115
(Declarations).....	115
Payroll_Load_Review [Sub ].....	115
Payroll_Load_Summary [Sub ].....	115
Payroll_ReviewEmpMacs .....	118
(Declarations).....	118
Payroll_LoadEmpReview [Sub ].....	118
Payroll_Review_EditTimeClockEntry [Sub ].....	120
Payroll_ShowEditEmployeeIcon [Sub ].....	121
Payroll_ShowEditTimeClockIcon [Sub ].....	121
Payroll_ViewEmployeeInfo [Sub ].....	121
Payroll_SaveEmplPayMac .....	123
(Declarations).....	123
Payroll_PrevEmployee [Sub ].....	123
Payroll_SaveEmployeePay [Sub ].....	123
Payroll_SelectSections.....	129
(Declarations).....	129
Payroll_Details_To_Review [Sub ].....	129
Payroll_SelectDetailsSection [Sub ].....	129
Payroll_SelectReviewSection [Sub ].....	129
Payroll_SelectSummarySection [Sub ].....	130
Payroll_Setup.....	132
(Declarations).....	132
Payroll_AddSelectedEmpl [Sub ].....	132
Payroll_EmpToPay_SelectAll [Sub ].....	132
Payroll_FilterEmployees [Sub ].....	132
Payroll_RemoveSelectedEmpl [Sub ].....	133
Payroll_SelToPay_SelectAll [Sub ].....	133
Payroll_Summary .....	136
(Declarations).....	136
Payroll_ConfirmCreate [Sub ].....	136
Payroll_LoadPayrollSummary [Sub ].....	136
Payroll_PrintPayStubs [Sub ].....	137
PopUp_Calendar.....	139
(Declarations).....	139
CalCol1 [Sub ].....	139
CalCol2 [Sub ].....	139
CalCol3 [Sub ].....	139
CalCol4 [Sub ].....	139
CalCol5 [Sub ].....	139
CalCol6 [Sub ].....	139
CalCol7 [Sub ].....	139
CalCol8 [Sub ].....	139
CalCol9 [Sub ].....	140
CalendarHide [Sub ].....	140
CalendarShow [Sub ].....	140
CalFormulaReplacement [Sub ].....	141
CheckForSheet [Sub ].....	141
CreateCalSht [Sub ].....	141
DayBtn1 [Sub ].....	143
DayBtn10 [Sub ].....	143
DayBtn11 [Sub ].....	144
DayBtn12 [Sub ].....	144
DayBtn13 [Sub ].....	144
DayBtn14 [Sub ].....	144
DayBtn15 [Sub ].....	144
DayBtn16 [Sub ].....	144

Table of Contents

DayBtn17 [Sub ]	144
DayBtn18 [Sub ]	145
DayBtn19 [Sub ]	145
DayBtn2 [Sub ]	145
DayBtn20 [Sub ]	145
DayBtn21 [Sub ]	145
DayBtn22 [Sub ]	145
DayBtn23 [Sub ]	145
DayBtn24 [Sub ]	146
DayBtn25 [Sub ]	146
DayBtn26 [Sub ]	146
DayBtn27 [Sub ]	146
DayBtn28 [Sub ]	146
DayBtn29 [Sub ]	146
DayBtn3 [Sub ]	146
DayBtn30 [Sub ]	147
DayBtn31 [Sub ]	147
DayBtn32 [Sub ]	147
DayBtn33 [Sub ]	147
DayBtn34 [Sub ]	147
DayBtn35 [Sub ]	147
DayBtn36 [Sub ]	148
DayBtn37 [Sub ]	148
DayBtn38 [Sub ]	148
DayBtn39 [Sub ]	148
DayBtn4 [Sub ]	148
DayBtn40 [Sub ]	148
DayBtn41 [Sub ]	148
DayBtn42 [Sub ]	149
DayBtn5 [Sub ]	149
DayBtn6 [Sub ]	149
DayBtn7 [Sub ]	149
DayBtn8 [Sub ]	149
DayBtn9 [Sub ]	149
GroupCal [Sub ]	149
MacroLinkRemover [Sub ]	150
NextMonth [Sub ]	151
PrevMonth [Sub ]	151
ShowSettings [Sub ]	151
UnGroupCal [Sub ]	152
Reports_ClearPrint	154
(Declarations)	154
Reports_Clear [Sub ]	154
Reports_PrintOut [Sub ]	154
Reports_EarningStmnt	156
(Declarations)	156
Reports_EarningsStatement [Sub ]	156
Reports_TimeclockRpt	160
(Declarations)	160
Report_TimelockReport [Sub ]	160
RightClick_Menu	162
(Declarations)	162
CreateLeaveEvent [Sub ]	162
CreateLeaveEvent10 [Sub ]	162
CreateLeaveEvent11 [Sub ]	162
CreateLeaveEvent12 [Sub ]	162
CreateLeaveEvent13 [Sub ]	162
CreateLeaveEvent14 [Sub ]	162
CreateLeaveEvent15 [Sub ]	162
CreateLeaveEvent16 [Sub ]	162
CreateLeaveEvent17 [Sub ]	162
CreateLeaveEvent18 [Sub ]	162
CreateLeaveEvent19 [Sub ]	163
CreateLeaveEvent20 [Sub ]	163
CreateLeaveEvent21 [Sub ]	163
CreateLeaveEvent22 [Sub ]	163



CreateLeaveEvent23 [Sub ] .....	163
CreateLeaveEvent7 [Sub ] .....	163
CreateLeaveEvent8 [Sub ] .....	163
CreateLeaveEvent9 [Sub ] .....	163
RightClick_CallEvents [Sub ] .....	163
RightClick_Cancel [Sub ] .....	164
Scheduling_Macs .....	166
(Declarations).....	166
Sched_AddComment_Event [Sub ] .....	166
Sched_AddComment_Holiday [Sub ] .....	167
Sched_ClearFilters [Sub ] .....	167
Sched_GoToEvent [Sub ] .....	167
Sched_MonthView [Sub ] .....	168
Sched_NextMonth [Sub ] .....	168
Sched_NextWeek [Sub ] .....	168
Sched_PrevMonth [Sub ] .....	168
Sched_PrevWeek [Sub ] .....	168
Sched_Refresh [Sub ] .....	169
Sched_ScrollToRow [Sub ] .....	169
Sched_ShowHideCapacity [Sub ] .....	169
Sched_ThisMonth [Sub ] .....	169
Sched_ThisWeek [Sub ] .....	169
Sched_WeekView [Sub ] .....	169
Time_Picker_Macros .....	171
(Declarations).....	171
AMPMSel [Sub ] .....	171
AMSel [Sub ] .....	171
DoneBtn [Sub ] .....	171
GrpHrs [Sub ] .....	171
GrpMin [Sub ] .....	171
GrpSet [Sub ] .....	171
HideHrSel [Sub ] .....	172
HideMinSel [Sub ] .....	172
HoursDisplay [Sub ] .....	172
MacroLinkRemover [Sub ] .....	173
MinDisplay [Sub ] .....	173
PMSel [Sub ] .....	174
ProtTP [Sub ] .....	174
Select1 [Sub ] .....	175
Select10 [Sub ] .....	175
Select11 [Sub ] .....	175
Select12 [Sub ] .....	175
Select2 [Sub ] .....	175
Select3 [Sub ] .....	176
Select4 [Sub ] .....	176
Select5 [Sub ] .....	176
Select6 [Sub ] .....	176
Select7 [Sub ] .....	176
Select8 [Sub ] .....	176
Select9 [Sub ] .....	176
SelectM00 [Sub ] .....	177
SelectM05 [Sub ] .....	177
SelectM10 [Sub ] .....	177
SelectM15 [Sub ] .....	177
SelectM20 [Sub ] .....	177
SelectM25 [Sub ] .....	177
SelectM30 [Sub ] .....	178
SelectM35 [Sub ] .....	178
SelectM40 [Sub ] .....	178
SelectM45 [Sub ] .....	178
SelectM50 [Sub ] .....	178
SelectM55 [Sub ] .....	178
Set10MinInt [Sub ] .....	178
Set15MinInt [Sub ] .....	179
Set5MinInt [Sub ] .....	179
SetHideShow [Sub ] .....	179

SetThm1 [Sub ] .....	179
SetThm2 [Sub ] .....	179
SetThm3 [Sub ] .....	179
SetThm4 [Sub ] .....	179
SetThm5 [Sub ] .....	180
TPShow [Sub ] .....	180
UnGrpHrs [Sub ] .....	180
UnGrpMin [Sub ] .....	180
UnGrpSet [Sub ] .....	181
UnProtTP [Sub ] .....	181
TimeClock_EditSaveEntry .....	183
(Declarations) .....	183
EditTimeClock_Calculate [Sub ] .....	183
EditTimeClock_Save [Sub ] .....	183
TimeClock_EditEntry [Sub ] .....	184
TimeClock_Macros .....	187
(Declarations) .....	187
Timeclock_AlreadyEntered [Sub ] .....	187
Timeclock_DisplayPicture [Sub ] .....	187
Timeclock_EnterID [Sub ] .....	187
Timeclock_IDNotFound [Sub ] .....	190
Timeclock_StartStop [Sub ] .....	191
TimeClock_RemoteSync .....	193
(Declarations) .....	193
TimeClock_SyncRemoteEntries [Sub ] .....	193
User_Login_Macros .....	196
(Declarations) .....	196
CheckUser [Sub ] .....	196
CloseWorkbook [Sub ] .....	197
HideBar [Sub ] .....	197
LogOut [Sub ] .....	197
ShowLoginForm [Sub ] .....	197
StartWorkbook [Sub ] .....	197
User_Rights_Macros .....	199
(Declarations) .....	199
User_SetAs_Admin [Sub ] .....	199
User_SetAs_Manager [Sub ] .....	199
User_SetAs_TimeClock [Sub ] .....	199
User_SetAs_User [Sub ] .....	199

1 Option Explicit

2  
3 Private Sub Worksheet\_Activate()  
4 SetOnkey (True)  
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 > 1 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
180 End Sub
```

**A**

Application, 9, 10  
Attach\_DisplayThumbnail, 11

**C**

Cal, 10  
CalendarHide, 10  
CalendarShow, 10  
Cells, 8-10  
CheckForSheet, 10  
ClearContents, 11  
Column, 8-10  
CountLarge, 10

**E**

EmpIDCard\_FieldChange, 11  
Empl\_Load, 8  
Empl\_SwitchHorizontalTabs, 10  
Empl\_TimeclockRefresh, 8  
EmployIDNotFound, 8  
EmpRow, 8  
Empty, 8, 9, 11  
EnableEvents, 10  
EntireRow, 8  
Explicit, 8

**F**

Find, 8  
FoundPayPeriod, 8

**H**

Hidden, 8

**I**

Intersect, 8-11  
IsNumeric, 8

**L**

LeaveID, 9  
LeaveListID, 9  
LeaveRow, 9

**M**

Max, 9

**O**

Offset, 8

**P**

PayItemRow, 9, 10  
PayPeriod, 8

**R**

Range, 8-11  
Row, 8-11

**S**

SchedRow, 8, 9  
SetOnkey, 8, 10  
Shape, 10  
Shapes, 10  
Sheet11, 9  
Sheet13, 9  
Sheet17, 9, 10  
Sheet2, 8  
Sheet4, 8  
SwitchVerticalTabs, 11

**T**

Target, 8-11

**U**

Undo, 10

**V**

Value, 8-11

**W**

WeekCol, 8, 9  
WeekRow, 8, 9  
Worksheet\_Activate, 8  
Worksheet\_Change, 8  
Worksheet\_Deactivate, 10  
Worksheet\_SelectionChange, 10  
WorksheetFunction, 9

**X**

xLUp, 9  
xlValues, 8  
xlWhole, 8

1 Option Explicit

2



**E**

Explicit, 13

1 Option Explicit

2

**E**

Explicit, [15](#)

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

**E**

Empty, [17](#)

**I**

Intersect, [17](#)

**R**

Range, [17](#)

**T**

Target, [17](#)

Timeclock\_EnterID, [17](#)

**V**

Value, [17](#)

**W**

Worksheet\_Change, [17](#)

1 Option Explicit

2

**E**

Explicit, [19](#)

```
1 Option Explicit
2
3 Private Sub TextBox1_Change()
4
5 End Sub
```



**E**

Explicit, [21](#)

**T**

TextBox1\_Change, [21](#)

```
1 Option Explicit
```

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

```
4 If Not Intersect(Target, Range("F3" )) Is Nothing Then 'Update To/From Dates on Payroll  
Change
```

```
5 Dim PayrollRow As Long
```

```
6 If Range("B5" ).Value = Empty Then Exit Sub
```

```
7 PayrollRow = Range("B5" ).Value 'Payroll Row
```

```
8 Range("H3" ).Value = Sheet18.Range("D" & PayrollRow).Value 'From Date
```

```
9 Range("J3" ).Value = Sheet18.Range("E" & PayrollRow).Value 'To Date
```

```
10 Reports_Clear
```

```
11 If Range("B3" ).Value = 1 Then Reports_EarningsStatement
```

```
12 If Range("B3" ).Value = 2 Then Report_TimelockReport
```

```
13 End If
```

```
14  
15 If Not Intersect(Target, Range("F4" )) Is Nothing Then 'Update To/From Dates on Pay  
Period Change
```

```
16 Dim PayPeriodRow As Long
```

```
17 If Range("B7" ).Value = Empty Then Exit Sub
```

```
18 PayPeriodRow = Range("B7" ).Value 'Payroll Row
```

```
19 Range("H3" ).Value = Sheet4.Range("E" & PayPeriodRow).Value 'From Date
```

```
20 Range("J3" ).Value = Sheet4.Range("F" & PayPeriodRow).Value 'To Date
```

```
21 Reports_Clear
```

```
22 If Range("B3" ).Value = 1 Then Reports_EarningsStatement
```

```
23 If Range("B3" ).Value = 2 Then Report_TimelockReport
```

```
24 End If
```

```
25  
26 'On Employee Change
```

```
27 If Not Intersect(Target, Range("H4" )) Is Nothing Then
```

```
28 Reports_Clear
```

```
29 If Range("B3" ).Value = 1 Then Reports_EarningsStatement
```

```
30 If Range("B3" ).Value = 2 Then Report_TimelockReport
```

```
31 End If
```

```
32 End Sub
```

```
33  
34 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

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

```
36 If Not Intersect(Target, Range("C6:C7" )) Is Nothing Then
```

```
37 Range("B4" ).Value = Target.Row
```

```
38 Reports_Clear
```

```
39 If Target.Row = 6 Then Reports_EarningsStatement
```

```
40 If Target.Row = 7 Then Report_TimelockReport
```

```
41 End If
```

```
42  
43 End Sub
```

**C**

CountLarge, [23](#)

**E**

Empty, [23](#)

Explicit, [23](#)

**I**

Intersect, [23](#)

**P**

PayPeriodRow, [23](#)

PayrollRow, [23](#)

**R**

Range, [23](#)

Report\_TimelockReport, [23](#)

Reports\_Clear, [23](#)

Reports\_EarningsStatement, [23](#)

Row, [23](#)

**S**

Sheet18, [23](#)

Sheet4, [23](#)

**T**

Target, [23](#)

**V**

Value, [23](#)

**W**

Worksheet\_Change, [23](#)

Worksheet\_SelectionChange, [23](#)

## Option Explicit

```

1 Private Sub Worksheet_Change(ByVal Target As Range)
2
3   If Target.CountLarge > 1 Then Exit Sub
4   If Not Intersect(Target, Range("F5" )) Is Nothing And Range("B4" ).Value <> Empty Then
5     'On Change of Pay Period and only on Pay Period Match
6     Range("J6" ).Value = Sheet4.Range("E" & Range("B4" ).Value + 208).Value 'From Pay
7     Date
8     Range("J8" ).Value = Sheet4.Range("F" & Range("B4" ).Value + 208).Value 'To Pay Date
9   End If
10
11   'On change of Select Payroll in Review Section, but not on Payroll Load
12   If Not Intersect(Target, Range("AL3" )) Is Nothing And Range("B1" ).Value = False Then
13     Payroll_Load_Review
14
15   'On change of Select Payroll in Summary Section, but not on Payroll Load
16   If Not Intersect(Target, Range("BK3" )) Is Nothing And Range("B1" ).Value = False Then
17     Payroll_Load_Summary
18
19   If Not Intersect(Target, Range("F7,F8,H7,H8" )) Is Nothing Then Payroll_FilterEmployees
20   'Run Payroll Filter on Change
21   'On Change of Payroll Date, Save Is On Existing Payroll, but not on Payroll Load
22   If Not Intersect(Target, Range("AM3,BD6" )) Is Nothing And Range("B1" ).Value = False And
23   Range("B3" ).Value <> Empty Then
24     Sheet18.Range("C" & Range("B3" ).Value).Value = Target.Value
25   End If
26 End Sub

```

```

27 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
28   If Target.CountLarge > 1 Then Exit Sub
29   Dim Cal As Shape 'Calendar
30   On Error Resume Next
31   Set Cal = Shapes("Calendar" )
32   On Error GoTo 0
33   If Cal Is Nothing Then 'Shape Deleted
34     Application.EnableEvents = False
35     On Error Resume Next
36     Application.Undo
37     On Error GoTo 0
38     Application.EnableEvents = True
39   End If
40   If Not Intersect(Target, Range("J6,J8,AM3,BD6" )) Is Nothing Then 'Change this to any
41   cell (or cells) you would like to have the Pop-Up Calendar Appear
42     CheckForSheet
43     CalendarShow
44   Else:
45     CheckForSheet
46     CalendarHide
47   End If
48
49   'On Selection of Employees To Add
50   On Error Resume Next
51   If Not Intersect(Target, Range("D11:D310,L11:L310" )) Is Nothing And Cells(Target.Row,
52   Target.Column + 1).Value <> Empty Then
53     If Target.Value = "p" Then
54       Target.Value = ""
55       If Target.Column = 4 Then Range("B7" ).Value = False Else: Range("B8" ).Value =
56       False 'Select All option set to False
57     End
58   Else: Target.Value = "p"

```

```
53     1 2 End If
54     End If
55
56     'On Selection of Employees To Review
57     If Not Intersect(Target, Range("AA9:AC310" )) Is Nothing And Range("AB" & Target.Row).
58     Value <> Empty Then
59         Range("B11" ).Value = Range("AB" & Target.Row).Value 'Set Employee ID To Review
60         Range("B12" ).Value = Target.Row 'Set Row To Review
61         Payroll_ShowEditEmployeeIcon
62         Payroll_LoadEmpReview 'Run Employee Refresh
63     Else:
64         Shapes("EditEmployeeBtn" ).Visible = msoFalse
65     End If
66
67     'On Selection of Time Clock Entry To Review
68     If Not Intersect(Target, Range("AE14:AJ44" )) Is Nothing And Range("AE" & Target.Row).
69     Value <> Empty Then
70         Range("B14" ).Value = Target.Row
71         Payroll_ShowEditTimeClockIcon
72     Else:
73         Range("B14" ).ClearContents 'Clear selected Row
74         Shapes("EditTimeClockBtn" ).Visible = msoFalse
75     End If
76 End Sub
```

## A

Application, [25](#)

## C

Cal, [25](#)

CalendarHide, [25](#)

CalendarShow, [25](#)

Cells, [25](#)

CheckForSheet, [25](#)

ClearContents, [26](#)

Column, [25](#)

CountLarge, [25](#)

## E

Empty, [25](#), [26](#)

EnableEvents, [25](#)

Explicit, [25](#)

## I

Intersect, [25](#), [26](#)

## M

msoFalse, [26](#)

## P

Payroll\_FilterEmployees, [25](#)

Payroll\_Load\_Review, [25](#)

Payroll\_Load\_Summary, [25](#)

Payroll\_LoadEmpReview, [26](#)

Payroll\_ShowEditEmployeeIcon, [26](#)

Payroll\_ShowEditTimeClockIcon, [26](#)

## R

Range, [25](#), [26](#)

Row, [25](#), [26](#)

## S

Shape, [25](#)

Shapes, [25](#), [26](#)

Sheet18, [25](#)

Sheet4, [25](#)

## T

Target, [25](#), [26](#)

## U

Undo, [25](#)

## V

Value, [25](#), [26](#)

Visible, [26](#)

## W

Worksheet\_Change, [25](#)

Worksheet\_SelectionChange, [25](#)

1 Option Explicit

2

**E**

Explicit, [28](#)



1 Option Explicit

2

**E**

Explicit, [30](#)

1 Option Explicit

2

**E**

Explicit, [32](#)

1 Option Explicit

2

**E**

Explicit, [34](#)

1 Option Explicit

2

**E**

Explicit, [36](#)



```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("D3:E3" )) Is Nothing And Range("B3" ).Value = False Then
5         Attach_Filter
6     End Sub
```

**A**

Attach\_Filter, [38](#)

**E**

Explicit, [38](#)

**I**

Intersect, [38](#)

**R**

Range, [38](#)

**T**

Target, [38](#)

**V**

Value, [38](#)

**W**

Worksheet\_Change, [38](#)

```

1 Option Explicit
2 Dim TimeFormat As Long
3
4 Private Sub Worksheet_Activate()
5     ActiveWindow.ScrollRow = 1
6 End Sub
7
8 Private Sub Worksheet_Change(ByVal Target As Range)
9     On Error Resume Next
10
11     'On Change Of Custom Field Time
12     If Not Intersect(Target, Range("G108:G111" )) Is Nothing Then
13         Dim FieldTypeNumber As Long
14         FieldTypeNumber = Application.WorksheetFunction.Match(Target.Value, Sheet4.Range(
15             "FieldTypes" ), 0)
16         If FieldTypeNumber <> 0 Then
17             If FieldTypeNumber <> 5 Then
18                 Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation.Delete 'Delete
19                 Any possible Data validation
20                 Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).NumberFormat = Sheet4.Range(
21                     "B" & 119 + FieldTypeNumber).Value
22             Else:
23                 With Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation
24                     .Delete
25                     .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween,
26                         Formula1:="=CustomList" & Target.Row - 107
27                     .IgnoreBlank = True
28                     .InCellDropdown = True
29                     .ShowInput = True
30                     .ShowError = True
31                 End With
32             End If
33         End If
34     End If
35
36     'On Change Of Existing (not New) Notifications, update table below, and not on
37     Notification load
38     If Not Intersect(Target, Range("E507:E509,D511,G507,G508,I508" )) Is Nothing And Range(
39         "B506" ).Value = False And Range("B508" ).Value = False Then
40         Dim NotifRow As Long
41         Dim NotifCol As Long
42         NotifRow = Range("B507" ).Value 'Get Table Row
43         NotifCol = Cells(Target.Row, Target.Column + 20).Value 'Get Table Column
44         Cells(NotifRow, NotifCol).Value = Target.Value 'Add Change to Table Below
45     End If
46
47     'On Change if Global Time Format, run Time format macro
48     If Not Intersect(Target, Range("J12" )) Is Nothing Then Admin_SetTotalFormat
49 End Sub
50
51 Private Sub Worksheet_Deactivate()
52     'Range("B3").Value = 4
53     'Admin_SwitchHorizontalTabs
54 End Sub
55
56 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
57     If Target.CountLarge > 1 Then Exit Sub
58     If Not Intersect(Target, Range("D4:K4" )) Is Nothing Then
59         Range("B3" ).Value = Target.Column
60         Range("F2" ).Select
61         Admin_SwitchHorizontalTabs
62     End If
63 End Sub

```

```
56     1 2 End If
57
58     'Set Checkbox for User To Send Emails or Display Notification Pop-ups (Make changes only
on existing users, not blank rows
59     If Not Intersect(Target, Range("I308:I404" )) Is Nothing And Range("E" & Target.Row).
Value <> Empty Then
60         Range("I308:I404" ).ClearContents 'Clear any Previous checkboxes
61         Target.Value = "ü"
62     End If
63     'Selection of Notifications to Load in cells above
64     If Not Intersect(Target, Range("D521:L541" )) Is Nothing And Range("D" & Target.Row).
Value <> Empty Then
65         Range("B507" ).Value = Target.Row
66         Notif_Load
67     End If
68 End Sub
```

## A

ActiveWindow, [40](#)  
Add, [40](#)  
Admin\_SetTotalFormat, [40](#)  
Admin\_SwitchHorizontalTabs, [40](#)  
AlertStyle, [40](#)  
Application, [40](#)

## C

Cells, [40](#)  
ClearContents, [41](#)  
Column, [40](#)  
CountLarge, [40](#)

## D

Delete, [40](#)

## E

Empty, [41](#)  
Explicit, [40](#)

## F

FieldTypeNumber, [40](#)  
Formula1, [40](#)

## I

IgnoreBlank, [40](#)  
InCellDropdown, [40](#)  
Intersect, [40](#), [41](#)

## M

Match, [40](#)

## N

Notif\_Load, [41](#)  
NotifCol, [40](#)  
NotifRow, [40](#)  
NumberFormat, [40](#)

## O

Operator, [40](#)

## R

Range, [40](#), [41](#)  
Row, [40](#), [41](#)

## S

ScrollRow, [40](#)  
Sheet1, [40](#)  
Sheet4, [40](#)  
ShowError, [40](#)  
ShowInput, [40](#)

## T

Target, [40](#), [41](#)  
TimeFormat, [40](#)

## V

Validation, [40](#)  
Value, [40](#), [41](#)

## W

Worksheet\_Activate, [40](#)  
Worksheet\_Change, [40](#)  
Worksheet\_Deactivate, [40](#)  
Worksheet\_SelectionChange, [40](#)  
WorksheetFunction, [40](#)

## X

xlBetween, [40](#)  
xlValidAlertStop, [40](#)  
xlValidateList, [40](#)

```

1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     SetOnkey (True)
5 End Sub
6
7 Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
8
9 End Sub
10
11 Private Sub Worksheet_Change(ByVal Target As Range)
12     Dim MapRng As Range
13     Dim FoundMapRng As Range
14     Dim CellAdd As String
15     Dim SelRow As Long
16     If Not Intersect(Target, Range("L13" )) Is Nothing Then
17         If Target.Value = "Yes" Then Range("14:21" ).EntireRow.Hidden = False Else: Range(
18             "14:21" ).EntireRow.Hidden = True
19         End If
20
21         'On Any Existing Event Change Update Event List Table AND Selected Item in List below
22         Events, But not on Event Load
23     If Not Intersect(Target, Range("D4:N21" )) Is Nothing And Range("B2" ).Value = False And
24         Range("B1" ).Value = False And IsNumeric(Cells(Target.Row, Target.Column + 23).Value) =
25         True And Cells(Target.Row, Target.Column + 23).Value <> Empty Then
26         Dim EventRow As Long
27         If Not Intersect(Target, Range("F13:G13" )) Is Nothing Then Sheet7.Range("AA" & Range(
28             "B3" ).Value).Value = Range("J10" ).Value + Range("L10" ).Value - Range("B11" ).Value
29         'Update Reminder On Date & Time
30         EventRow = Range("B3" ).Value 'Event List Row
31         If Target.Column = 4 Then 'Reminders & Recurring Options
32             If InStr(Target.Value, "p" ) > 0 Then
33                 Sheet7.Cells(EventRow, Cells(Target.Row, Target.Column + 23).Value).Value =
34                     "Yes"
35                 Sheet7.Range("AA" & EventRow).Value = Range("J10" ).Value + Range("L10" ).Value
36                 - Range("B11" ).Value 'Set Reminder On Date & Time
37             Else: Sheet7.Cells(EventRow, Cells(Target.Row, Target.Column + 23).Value).Value =
38                 "No"
39             End If
40         Else: 'All Other fields
41             Sheet7.Cells(EventRow, Cells(Target.Row, Target.Column + 23).Value).Value = Target.
42             Value
43         End If
44         Sheet7.Range(EventRow & ":" & EventRow).WrapText = False
45         'Update Table Below
46         CellAdd = Target.Address
47         SelRow = Range("B10" ).Value
48         Set MapRng = Sheet5.Range("EventDataMap" )
49         Set FoundMapRng = MapRng.Find(CellAdd, , xlValues, xlWhole)
50         If Not FoundMapRng Is Nothing Then Cells(SelRow, FoundMapRng.Column).Value = Sheet7.
51         Cells(Range("B3" ).Value, Cells(Target.Row, Target.Column + 23).Value).Value
52     End If
53
54     'On change of employee, update Employee ID and Email Address, but not on Event Load
55     If Not Intersect(Target, Range("K4" )) Is Nothing And Range("B5" ).Value <> Empty And
56     Range("B1" ).Value = False Then
57         Range("F15" ).Value = Sheet2.Range("M" & Range("B5" ).Value).Value 'Add Employee
58         Email
59         Range("B6" ).Value = Sheet2.Range("A" & Range("B5" ).Value).Value
60         If Range("B3" ).Value <> Empty Then Sheet7.Range("E" & Range("B3" ).Value).Value =
61         Range("B6" ).Value 'Update Employee ID on Employee Change

```

```

1 2
48 End If
49 'Add in Default Notifications but not on Event Load and only for New Events
50 If Not Intersect(Target, Range("H4" )) Is Nothing And Range("B1" ).Value = False And Range
    ("B2" ).Value = True Then Event_AddNotifDefaults
51
52 'Run Table Filter on change but not when text contains the word "Enter"
53 If Not Intersect(Target, Range("D26,E26,G26" )) Is Nothing And InStr(Target.Value,
    "Enter" ) = 0 Then Events_FilterTable
54
55 'On Any Change of Event Type, Show All Day Button for Training and Leave Type Events
56 If Not Intersect(Target, Range("H4" )) Is Nothing Then
57
58
59     If Range("H4" ).Value = "Leave" Or Range("H4" ).Value = "Training" Then Shapes(
        "AllDayBtn" ).Visible = msoCTrue Else: Shapes("AllDayBtn" ).Visible = msoFalse
60 End If
61
62 End Sub
63
64 Private Sub Worksheet_Deactivate()
65     Dim TP As Shape
66     On Error Resume Next
67     Set TP = Shapes("TimePickGrp" )
68     On Error GoTo 0
69     If TP Is Nothing Then 'Shape Deleted
70         Application.EnableEvents = False
71         On Error Resume Next
72         Application.Undo
73         Application.EnableEvents = True
74     End If
75 End Sub
76
77 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
78     If Target.CountLarge > 1 Then Exit Sub
79     'For Pop-up Calendar
80     Dim Cal As Shape 'Calendar
81     On Error Resume Next
82     Set Cal = Shapes("Calendar" )
83     On Error GoTo 0
84     If Cal Is Nothing Then 'Shape Deleted
85         Application.EnableEvents = False
86         On Error Resume Next
87         Application.Undo
88         On Error GoTo 0
89         Application.EnableEvents = True
90     End If
91     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
92         CheckForSheet
93         CalendarShow
94     Else:
95         CheckForSheet
96         CalendarHide
97     End If
98
99     'For Pop-up Date Picker
100    Dim TP As Shape 'Time Picker
101    On Error Resume Next
102    Set TP = Shapes("TimePickGrp" )
103    On Error GoTo 0
104    If TP Is Nothing Then 'Shape Deleted

```

1 2

```

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

```



**A**

ActiveCell, 45  
Address, 43  
Application, 44, 45

**C**

Cal, 44  
CalendarHide, 44  
CalendarShow, 44  
Cancel, 43  
CellAdd, 43  
Cells, 43  
Characters, 45  
CheckForSheet, 44  
Column, 43  
CountLarge, 44

**E**

Empty, 43, 45  
EnableEvents, 44, 45  
EntireRow, 43  
Event\_AddNotifDefaults, 44  
Event\_Load, 45  
EventRow, 43  
Events\_FilterTable, 44  
Explicit, 43

**F**

Find, 43  
Font, 45  
FoundMapRng, 43

**H**

Hidden, 43

**I**

InStr, 43, 44  
Intersect, 43-45  
IsNumeric, 43

**L**

Length, 45

**M**

MapRng, 43  
msoCTrue, 44, 45  
msoFalse, 44, 45

**N**

Name, 45

**R**

Range, 43-45  
Row, 43, 45

**S**

SelRow, 43  
SetOnkey, 43, 45  
Shape, 44  
Shapes, 44, 45  
Sheet2, 43  
Sheet5, 43, 45  
Sheet7, 43  
Start, 45

**T**

Target, 43-45  
TP, 44  
TPShow, 45

**U**

Undo, 44, 45

**V**

Value, 43-45  
Visible, 44, 45

**W**

Worksheet\_Activate, 43  
Worksheet\_BeforeDoubleClick, 43  
Worksheet\_Change, 43  
Worksheet\_Deactivate, 44  
Worksheet\_SelectionChange, 44  
WrapText, 43

**X**

xlValues, 43  
xlWhole, 43

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Not Intersect(Target, Range("A4:AG999" )) Is Nothing And Range("A" & Target.Row).Value <> Empty Then
5         Range("BA1" ).Value = Target.Row
6     Else: Range("BA1" ).ClearContents
7     End If
8 End Sub
```

**C**

ClearContents, [47](#)

**E**

Empty, [47](#)

Explicit, [47](#)

**I**

Intersect, [47](#)

**R**

Range, [47](#)

Row, [47](#)

**T**

Target, [47](#)

**V**

Value, [47](#)

**W**

Worksheet\_SelectionChange, [47](#)

1 Option Explicit

2

**E**

Explicit, [49](#)

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

## A

ActiveSheet, [51](#)  
ActiveWindow, [51](#)  
AddCol, [51](#)  
Address, [51](#)  
Application, [51](#)  
AutoFilter, [51](#)  
AutoFilterMode, [51](#)

## C

Cells, [51](#)  
ClearComments, [51](#)  
Column, [51](#)  
Count, [51](#)  
Criteria1, [51](#)

## D

DisplayCommentIndicator, [51](#)

## E

Empty, [51](#)  
Explicit, [51](#)

## F

Field, [51](#)

## I

Intersect, [51](#)

## R

Range, [51](#)  
Row, [51](#)

## S

Sched\_AddComment\_Event, [51](#)  
Sched\_AddComment\_Holiday, [51](#)  
Sched\_MonthView, [51](#)  
Sched\_Refresh, [51](#)  
Sched\_ScrollToRow, [51](#)  
Sched\_WeekView, [51](#)  
ScrollRow, [51](#)  
ShowAllData, [51](#)

## T

Target, [51](#)

## V

Value, [51](#)

## W

Worksheet\_Activate, [51](#)  
Worksheet\_Change, [51](#)  
Worksheet\_SelectionChange, [51](#)

## X

xlCommentIndicatorOnly, [51](#)

1 Option Explicit

2



**E**

Explicit, [53](#)

```
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 End Sub
13
14 Private Sub Workbook_BeforeClose(Cancel As Boolean)
15     On Error Resume Next
16     Application.CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
17     Application.CommandBars("Cell" ).Controls("Go To Event" ).Delete
18     Application.CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
19     On Error GoTo 0
20 End Sub
21
22 Private Sub Workbook_Deactivate()
23     SetOnkey (False)
24     On Error Resume Next
25     Application.CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
26     Application.CommandBars("Cell" ).Controls("Go To Event" ).Delete
27     Application.CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
28     On Error GoTo 0
29 End Sub
30
31 Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel
32     As Boolean)
33     On Error Resume Next
34     Application.CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
35     Application.CommandBars("Cell" ).Controls("Go To Event" ).Delete
36     Application.CommandBars("Cell" ).Controls("Edit Time Clock Entry" ).Delete
37     On Error GoTo 0
38     If ActiveSheet.CodeName = "Sheet8" Then RightClick_CallEvents
39 End Sub
```

## A

ActiveSheet, [55](#)  
Application, [55](#)

## C

Cancel, [55](#)  
CodeName, [55](#)  
CommandBars, [55](#)  
Controls, [55](#)

## D

Delete, [55](#)

## E

Explicit, [55](#)

## R

Range, [55](#)  
RightClick\_CallEvents, [55](#)

## S

SetOnkey, [55](#)  
Sh, [55](#)

## T

Target, [55](#)

## W

Workbook\_Activate, [55](#)  
Workbook\_BeforeClose, [55](#)  
Workbook\_Deactivate, [55](#)  
Workbook\_SheetBeforeRightClick, [55](#)

```
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, [57](#)  
ClsBtn\_Click, [57](#)  
CreateBtn\_Click, [57](#)  
CreatePayPeriod, [57](#)  
CreatePayPeriods, [57](#)

## E

Explicit, [57](#)

## H

Hide, [57](#)

```
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, [59](#), [60](#)  
BreakIn\_AfterUpdate, [59](#)  
BreakInNow\_MouseDown, [59](#)  
BreakOut, [59](#), [60](#)  
BreakOut\_AfterUpdate, [59](#)  
BreakOutNow\_MouseDown, [59](#)  
Button, [59](#)

## C

CancelBtn\_MouseDown, [59](#)  
ClockIn, [59](#), [60](#)  
ClockIn\_AfterUpdate, [59](#)  
ClockInNow\_MouseDown, [59](#)  
ClockOut, [59](#), [60](#)  
ClockOut\_AfterUpdate, [59](#)  
ClockOutNow\_MouseDown, [59](#)

## E

EditTimeClock, [59](#)  
EditTimeClock\_Calculate, [59](#)  
EditTimeClock\_Save, [59](#)  
EmplName, [60](#)  
Explicit, [59](#)

## F

Format, [59](#), [60](#)

## H

Hide, [59](#)

## N

Now, [59](#)

## O

OkBtn\_Click, [59](#)

## R

Range, [60](#)

## S

Sheet8, [60](#)  
Shift, [59](#)

## T

text, [60](#)

## U

UserForm\_Activate, [60](#)

## V

Value, [59](#), [60](#)

## X

X, [59](#)

## Y

Y, [59](#)



```
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 Private Sub UserForm_Initialize()  
15     HideBar Me 'Remove Border and Title Bar  
16 End Sub
```

## C

CheckUser, [62](#)  
CloseBtn\_Click, [62](#)

## E

Explicit, [62](#)

## H

Hide, [62](#)  
HideBar, [62](#)

## L

LoginForm, [62](#)

## O

OkBtn\_Click, [62](#)

## U

UserForm\_Initialize, [62](#)

```
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, [64](#)  
Application, [64](#)

## C

Cells, [64](#), [65](#)  
ClearContents, [64](#)

## E

Empty, [64](#), [65](#)  
Explicit, [64](#)

## F

FileDialog, [64](#)  
FileFldr, [64](#)  
Filters, [64](#)

## L

LastRow, [64](#)

## M

MsgBox, [64](#), [65](#)  
msoCTrue, [64](#), [65](#)  
msoFalse, [64](#), [65](#)  
msoFileDialogFilePicker, [64](#)

## N

NoSelection, [64](#)  
Notif\_AddEmailAttach, [64](#)  
Notif\_AddNew, [64](#)  
Notif\_CancelNew, [64](#)  
Notif\_Delete, [64](#)  
Notif\_Load, [64](#)  
Notif\_Save, [64](#)  
NotifCol, [64](#), [65](#)  
NotifRow, [64](#), [65](#)

## R

Range, [64](#), [65](#)  
Row, [64](#), [65](#)

## S

SelectedItems, [64](#)  
Shapes, [64](#), [65](#)  
Sheet4, [64](#), [65](#)  
Show, [64](#)

## T

Title, [64](#)

## V

Value, [64](#), [65](#)  
vbNo, [64](#)  
vbYesNo, [64](#)  
Visible, [64](#), [65](#)

## X

xlUp, [64](#), [65](#)

```

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 BrowseEmployeeFolder()
16     Dim EmpFldr As FileDialog
17     Dim strPath As String
18     Set EmpFldr = Application.FileDialog(msoFileDialogFolderPicker)
19     With EmpFldr
20         .Title = "Select a Folder"
21         .AllowMultiSelect = False
22         If .Show <> -1 Then GoTo NoSelection
23         Sheet4.Range("EmplAttachFolder" ).Value = .SelectedItems(1)
24     End With
25     NoSelection:
26 End Sub
27
28 Sub BrowseRemoteTimeClockFolder()
29     Dim EmpFldr As FileDialog
30     Dim strPath As String
31     Set EmpFldr = Application.FileDialog(msoFileDialogFolderPicker)
32     With EmpFldr
33         .Title = "Select a Folder"
34         .AllowMultiSelect = False
35         If .Show <> -1 Then GoTo NoSelection
36         Sheet4.Range("RemoteTimeClockFolder" ).Value = .SelectedItems(1)
37     End With
38     NoSelection:
39 End Sub
40
41 Sub CreatePayPeriods()
42     Dim PayrollQty As Long
43     Dim StartRow As Long
44     Dim FormulaRow As Long
45     Dim StartDate As Date
46     Dim NextDate As Date
47     Dim Frequency As String
48     With Sheet4
49         If .Range("B4" ).Value = Empty Or .Range("B5" ).Value = Empty Or .Range("B6" ).Value =
50             Empty Then
51             MsgBox "Please make sure that all fields have been filled in before Creating Pay
52                 Periods"
53             Exit Sub
54         End If
55         CreatePayPeriod.Hide
56         Frequency = .Range("B4" ).Value 'Frequency
57         StartDate = .Range("B5" ).Value 'Start Date
58         PayrollQty = .Range("B6" ).Value 'Payroll Qty
59
60         Select Case Frequency
61             Case Is = "Weekly"
62                 FormulaRow = 15
63             Case Is = "Bi-Weekly"

```

1 2 3

```
1 2 3
60     FormulaRow = 17
61     Case Is = "Semi-Monthly"
62         FormulaRow = 19
63     Case Is = "Monthly"
64         FormulaRow = 21
65     End Select
66
67     If .Range("B7" ).Value = True Then 'New List
68         .Range("D209:F304" ).ClearContents
69         StartRow = 209
70     Else:
71         StartRow = .Range("D304" ).End(xlUp).Row + 1 'First Available Row
72     End If
73     .Range("E" & StartRow).Value = StartDate
74
75     .Range("B" & FormulaRow).Copy
76     .Range("F" & StartRow & ":F" & StartRow + PayrollQty - 1).PasteSpecial
77     xlPasteFormulas
78     .Range("A" & FormulaRow + 1).Copy
79     .Range("E" & StartRow + 1 & ":E" & StartRow + PayrollQty - 1).PasteSpecial
80     xlPasteFormulas
81     .Range("B14" ).Copy
82     .Range("D" & StartRow & ":D" & StartRow + PayrollQty - 1).PasteSpecial
83     xlPasteFormulas 'Create Pay Period Names
84
85 End With
86 Application.CutCopyMode = False
87 End Sub
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

## A

Admin\_SetTotalFormat, [67](#)  
AllowMultiSelect, [67](#)  
Application, [67](#), [68](#)

## B

BrowseEmployeeFolder, [67](#)  
BrowseRemoteTimeClockFolder, [67](#)

## C

ClearContents, [68](#)  
Copy, [68](#)  
CreatePayPeriod, [67](#), [68](#)  
CreatePayPeriods, [67](#)  
CutCopyMode, [68](#)

## E

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

## F

FileDialog, [67](#)  
FormulaRow, [67](#), [68](#)  
Frequency, [67](#)

## H

Hide, [67](#)

## M

MsgBox, [67](#)  
msoFileDialogFolderPicker, [67](#)

## N

NextDate, [67](#)  
NoSelection, [67](#)  
NumberFormat, [67](#)

## P

PasteSpecial, [68](#)  
PayrollQty, [67](#), [68](#)

## R

Range, [67](#), [68](#)  
Row, [68](#)

## S

SelectedItems, [67](#)  
Sheet1, [67](#)  
Sheet15, [67](#)  
Sheet16, [67](#)  
Sheet4, [67](#)  
Show, [67](#), [68](#)  
ShowPayPeriodForm, [68](#)  
StartDate, [67](#), [68](#)  
StartRow, [67](#), [68](#)  
strPath, [67](#)

## T

TimeInteger, [67](#)  
Title, [67](#)

## V

Value, [67](#), [68](#)

## X

xlPasteFormulas, [68](#)  
xlUp, [68](#)



```
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, 70

**E**

EntireRow, 70

Explicit, 70

**F**

FirstRow, 70

**H**

Hidden, 70

**M**

msoCTrue, 70

msoFalse, 70

**R**

Range, 70

ResetCalc, 70

**S**

SelCol, 70

Shapes, 70

Sheet4, 70

StopCalc, 70

**V**

Value, 70

Visible, 70

```

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 End Function
47
48 Sub SetOnkey(ByVal state As Boolean)
49   'Compiled By Randy Austin
50   'Workbook Provided By www.ExcelForFreelancers.com
51   If state Then
52     With Application
53       .OnKey "{TAB}" , "'TabRange xlNext'"
54       .OnKey "~" , "'TabRange xlNext'"
55       .OnKey "{ENTER}" , "'TabRange xlNext'"
56       .OnKey "{RIGHT}" , "'TabRange xlNext'"
57       .OnKey "{LEFT}" , "'TabRange xlPrevious'"
58       .OnKey "+{TAB}" , "'TabRange xlPrevious'"
59       .OnKey "{DOWN}" , "'UpOrDownArrow xlDown'"
60       .OnKey "{UP}" , "'UpOrDownArrow xlUp'"
61     End With
62   Else

```

```

1 2
55 'reset keys
56 With Application
57     .OnKey "{ENTER}"
58     .OnKey "{TAB}"
59     .OnKey "~"
60     .OnKey "+{TAB}"
61     .OnKey "{RIGHT}"
62     .OnKey "{LEFT}"
63     .OnKey "{DOWN}"
64     .OnKey "{UP}"
65 End With
66 End If
67 End Sub
68
69 Sub TabRange(Optional iDirection As Integer = xlNext)
70
71     Dim vTabOrder As Variant, m As Variant
72     Dim lItems As Long, iAdjust As Long
73     On Error GoTo ExitSub
74     '--get the tab order from shared function
75     vTabOrder = GetTabOrder
76     lItems = UBound(vTabOrder) - LBound(vTabOrder) + 1
77
78     On Error Resume Next
79     m = Application.Match(ActiveCell.Address(0, 0), vTabOrder, False)
80
81
82     '--if activecell is not in Tab Order return to the first cell
83     If IsError(m) Then
84         m = 1
85     Else
86         '--get adjustment to index
87         iAdjust = IIf(iDirection = xlPrevious, -1, 1)
88
89         '--calculate new index wrapping around list
90         m = (m + lItems + iAdjust - 1) Mod lItems + 1
91     End If
92
93     '--select cell adjusting for Option Base 0 or 1
94     Application.EnableEvents = False
95     Range(vTabOrder(m + (LBound(vTabOrder) = 0))).Select
96
97     ExitSub:
98     Application.EnableEvents = True
99
100
101 End Sub
102
103 Sub UpOrDownArrow(Optional iDirection As Integer = xlUp)
104
105     Dim vTabOrder As Variant
106     Dim lRowClosest As Long, lRowTest As Long
107     Dim i As Long, iSign As Integer
108
109     Dim sActiveCol As String
110     Dim bFound As Boolean
111
112     '--get the tab order from shared function
113     vTabOrder = GetTabOrder
114     On Error Resume Next
115     If vTabOrder = Empty Then Exit Sub

```

1

```
1
116 '--find TabCells in same column as ActiveCell in iDirection
117 '-- rTest will include ActiveCell
118
119 sActiveCol = GetColLtr(ActiveCell.Address(0, 0))
120
121 iSign = IIf(iDirection = xlDown, -1, 1)
122 lRowClosest = IIf(iDirection = xlDown, Rows.Count + 1, 0)
123
124 For i = LBound(vTabOrder) To UBound(vTabOrder)
125     If GetColLtr(CStr(vTabOrder(i))) = sActiveCol Then
126         lRowTest = Range(CStr(vTabOrder(i))).Row
127
128         '--find closest cell to ActiveCell in rTest
129         If iSign * lRowTest > iSign * lRowClosest And _
130             iSign * lRowTest < iSign * ActiveCell.Row Then
131             '--at least one cell in iDirection of same column
132             bFound = True
133             lRowClosest = lRowTest
134         End If
135     End If
136 Next i
137
138 If bFound Then
139     Application.EnableEvents = False
140     Cells(lRowClosest, ActiveCell.Column).Select
141     Application.EnableEvents = True
142 End If
143 End Sub
```

→, 74

## A

ActiveCell, 72-74  
ActiveSheet, 72  
Address, 73, 74  
Application, 72-74  
Array, 72

## B

bFound, 73, 74

## C

Cells, 74  
CodeName, 72  
Column, 74  
Count, 74

## D

do\_nothing, 72

## E

Empty, 73  
EnableEvents, 73, 74  
ExitSub, 73

## G

GetColLtr, 72, 74  
GetTabOrder, 72, 73

## I

i, 73, 74  
iAdjust, 73  
iDirection, 73, 74  
If, 73, 74  
iPos, 72  
IsError, 73  
iSign, 73, 74  
IsNumeric, 72

## L

LBound, 73, 74  
lItems, 73  
lRowClosest, 73, 74  
lRowTest, 73, 74

## M

m, 73  
Match, 73  
Mid, 72

## O

OnKey, 72, 73

## R

Range, 73, 74  
Row, 72, 74  
Rows, 74

## S

sActiveCol, 73, 74  
sAddr, 72  
SetOnkey, 72  
state, 72  
sTest, 72

## T

TabRange, 73

## U

UBound, 73, 74  
UpOrDownArrow, 73

## V

vTabOrder, 73, 74

## X

xlDown, 74  
xlNext, 73  
xlPrevious, 73  
xlUp, 73

```
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, [76](#)

**C**

Calculation, [76](#)

**E**

EnableEvents, [76](#)

Explicit, [76](#)

**R**

ResetCalc, [76](#)

**S**

ScreenUpdating, [76](#)

StopCalc, [76](#)

**X**

xlCalculationAutomatic, [76](#)

xlCalculationManual, [76](#)



```

1  Option Explicit
2
3  Sub Attach_Add()
4      Dim FileFldr As FileDialog
5      Dim FileName, OrigFilePath, FileType, DestFileFolder, DestFilePath, EmpID, EmpName As String
6      Dim LastAttRow As Long
7      DestFileFolder = Sheet4.Range("EmplAttachFolder").Value
8      If DestFileFolder = "" Then 'If user has not added a main folder yet
9          MsgBox "Before attaching any files for this Employee, please select a dedicated" &
10             vbCrLf & "folder to store the individual Employee Attachments"
11         Exit Sub
12     End If
13     EmpID = Sheet1.Range("J2").Value 'Employee ID
14     EmpName = Replace(Replace(Sheet1.Range("F2").Value, ",", "_"), " ", "") 'Employee Name
15     Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
16     With FileFldr
17         .Title = "Select file to attach"
18         .Filters.Add "All Files", "*.*"
19         If .Show <> -1 Then GoTo NoSelection
20         OrigFilePath = .SelectedItems(1)
21         FileName = Dir(OrigFilePath)
22
23         FileType = Right(FileName, 5 - InStr(1, Right(FileName, 5), "."))
24     End With
25
26     'Check if Employee Folder exists, if not, add it.
27     If Len(Dir(DestFileFolder & "\" & EmpName, vbDirectory)) = 0 Then
28         On Error GoTo BadFolder
29         MkDir DestFileFolder & "\" & EmpName
30     End If
31     DestFilePath = DestFileFolder & "\" & EmpName & "\" & FileName
32     FileCopy OrigFilePath, DestFilePath 'Copy user selected file into appropriate Employee folder
33     'Kill (OrigFilePath) 'If you want to delete the original file in the original location
34
35     With Sheet3
36         LastAttRow = .Range("D9999").End(xlUp).Row + 1 'First Avail. Attachment Row
37         .Range("D" & LastAttRow).Value = EmpID 'EmployeeID
38         .Range("E" & LastAttRow).Value = Sheet1.Range("F2").Value 'Employee Name
39         .Range("F" & LastAttRow).Value = FileName
40         .Range("G" & LastAttRow).Value = FileType
41         .Range("H" & LastAttRow).Value = DestFilePath
42         .Range("I" & LastAttRow).Value = Application.UserName
43         .Range("J" & LastAttRow).Value = Now
44         .Range("K" & LastAttRow).Value = "=Row()"
45     End With
46
47     Attach_Refresh 'Refresh List To Display New Attachment
48     Sheet1.Range("E19").Select 'Select First Item to Display Thumbnail (if any)
49     Attach_DisplayThumbnail
50     NoSelection:
51     Exit Sub
52     BadFolder:
53     MsgBox "Please select a correct folder to add attachments"
54     Sheet4.Activate
55     BrowseEmployeeFolder
56 End Sub
57 Sub Attach_Delete()

```

1

```

1
58 Dim EmpID, AttRow As Long
59 If Sheet1.Range("J" & ActiveCell.Row).Value = Empty Then
60     MsgBox "Please select an Attachment to delete first"
61     Exit Sub
62 End If
63 On Error Resume Next
64 Sheet1.Shapes("AttachThumb" ).Delete
65 On Error GoTo 0
66
67 AttRow = Sheet1.Range("J" & ActiveCell.Row).Value 'Attachement Row
68 Sheet3.Range(AttRow & ":" & AttRow).EntireRow.Delete
69 Attach_Refresh
70 Sheet1.Range("E19" ).Select
71 Attach_DisplayThumbnail
72 End Sub
73
74 Sub Attach_DisplayThumbnail()
75     Dim AttRow As Long
76     Dim AttTableRow As Long
77     Dim FileType, FilePath, ThumbType As String
78     With Sheet1
79         On Error Resume Next
80         .Shapes("AttachThumb" ).Delete
81         On Error GoTo 0
82         If .Range("B7" ).Value = Empty Then Exit Sub 'Exit on No AttachmentRow
83         AttRow = .Range("B7" ).Value 'Set Attachment Row
84         FileType = LCase(.Range("F" & AttRow).Value)
85         AttTableRow = .Range("J" & AttRow).Value 'Attachement Table Row
86         FilePath = .Range("G" & AttRow).Value 'Full File Path
87         If FileType = "jpg" Or FileType = "png" Or FileType = "gif" Or FileType = "jpeg" Or
88             FileType = "bmp" Then 'Picture Types
89             On Error GoTo NoAttach
90             .Pictures.Insert(FilePath).Name = "AttachThumb"
91             With .Shapes("AttachThumb" )
92                 .Visible = msoCTrue
93                 .LockAspectRatio = True
94                 .Left = Sheet1.Range("J19" ).Left
95                 .Top = Sheet1.Range("J19" ).Top
96                 If .Width > .Height Then .Width = 90 Else: .Height = 80
97                 .IncrementLeft 30
98                 .IncrementTop 2
99             End With
100             'On Word, Excel or PDF Section
101         ElseIf FileType = "docx" Or FileType = "pdf" Or FileType = "doc" Then
102             On Error Resume Next
103             .OLEObjects.Add(FileName:=FilePath, Link:=False, DisplayAsIcon:=False).Name =
104                 "AttachThumb"
105             On Error GoTo NoAttach
106             With .Shapes("AttachThumb" )
107                 .LockAspectRatio = True
108                 .Left = Sheet1.Range("L4" ).Left
109                 .Top = Sheet1.Range("L4" ).Top
110                 If .Width > .Height Then .Width = 300 Else: .Height = 500
111                 .IncrementLeft 35
112                 .IncrementTop 2
113                 .Visible = msoCTrue
114             End With
115         End If
116     End With

```

```

1
117 NoAttach:
118 End Sub
119 Sub Attach_Filter()
120 Dim LastAttRow As Long
121 With Sheet3
122 .Activate
123 .Range("AA2:AH2").ClearContents 'Clear Old Criteria
124 If .Range("D3").Value <> "Enter ID:" And .Range("D3").Value <> Empty Then .Range(
125 "AA2").Value = .Range("D3").Value Else: .Range("AA2").ClearContents 'Employee ID
126 If .Range("E3").Value <> "Enter Name:" And .Range("E3").Value <> Empty Then .Range(
127 "AB2").Value = "*" & .Range("E3").Value & "*" Else: .Range("AB2").ClearContents
128 'Employee Name
129 LastAttRow = .Range("D9999").End(xlUp).Row
130 If LastAttRow < 5 Then GoTo NoData
131 .Range("D4:K" & LastAttRow).AdvancedFilter xlFilterInPlace, CriteriaRange:=.Range(
132 "AA1:AH2"), Unique:=True
133 NoData:
134 End With
135 End Sub
136
137 Sub Attach_FilterClear()
138 With Sheet3
139 On Error Resume Next
140 .ShowAllData
141 On Error GoTo 0
142 .Range("B3").Value = True 'Set Filter to True
143 .Range("D3").Value = "Enter ID:"
144 .Range("E3").Value = "Enter Name:"
145 .Range("B3").Value = False 'Set Filter to False
146 .Range("D5:D9999").EntireRow.Hidden = False
147 End With
148 End Sub
149
150 Sub Attach_Open()
151 With Sheet1
152 Dim AttRow As Long
153 If .Range("B7").Value = Empty Then
154 MsgBox "Please select an attachment to open"
155 Exit Sub
156 End If
157 AttRow = .Range("B7").Value 'Attachment Row
158 ThisWorkbook.FollowHyperlink (.Range("G" & AttRow).Value)
159 End With
160 End Sub
161
162 Sub Attach_Refresh()
163 Dim EmpID, AttRow, LastAttRow, LastFiltRow, FiltRow As Long
164
165 EmpID = Sheet1.Range("J2").Value 'Empoyee ID
166 Sheet1.Range("E19:J24").ClearContents 'Clear Old Attachemets
167 AttRow = 19 'Set Starting Attachments Row
168 With Sheet3
169 .Range("AA2:AH2,AA4:AH9999").ClearContents 'Clear Prev Adv. Filter
170 .Range("AA2").Value = EmpID 'Set Empl. ID as Filter
171 LastAttRow = .Range("D9999").End(xlUp).Row
172 If LastAttRow < 5 Then GoTo NoData
173 .Range("D4:K" & LastAttRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
174 "AA1:AH2"), CopytoRange:=.Range("AA3:AG3"), Unique:=True
175 LastFiltRow = .Range("AA9999").End(xlUp).Row 'Last Filtered Row
176 If LastFiltRow < 4 Then Exit Sub 'If no Attach. found then Exit
177 For FiltRow = LastFiltRow To 4 Step -1 'Use Reverse For next to put most recent
178 Attach. First

```

1 2 3

```
172     1 2 3 Sheet1.Range("E" & AttRow & ":J" & AttRow).Value = .Range("AB" & FiltRow & ":AG"  
173         & FiltRow).Value 'Copy Over Attachment Info  
174         AttRow = AttRow + 1  
175         If AttRow = 25 Then Exit Sub 'Exit If Table is Full  
176     Next FiltRow  
177 End With  
178 NoData:  
179 End Sub  
180  
181  
182 Sub Attach_ShowAll()  
183     Sheet3.Range("D3" ).Value = Sheet1.Range("J2" ).Value 'Employee ID  
184 End Sub
```

**A**

Activate, [78, 80](#)  
ActiveCell, [79](#)  
Add, [78, 79](#)  
AdvancedFilter, [80](#)  
Application, [78](#)  
Attach\_Add, [78](#)  
Attach\_Delete, [78](#)  
Attach\_DisplayThumbnail, [78, 79](#)  
Attach\_Filter, [80](#)  
Attach\_FilterClear, [80](#)  
Attach\_Open, [80](#)  
Attach\_Refresh, [78-80](#)  
Attach\_ShowAll, [81](#)  
AttRow, [79-81](#)  
AttTableRow, [79](#)

**B**

BadFolder, [78](#)  
BrowseEmployeeFolder, [78](#)

**C**

ClearContents, [80](#)  
CopytoRange, [80](#)  
CriteriaRange, [80](#)

**D**

Delete, [79](#)  
DestFileFolder, [78](#)  
DestFilePath, [78](#)  
Dir, [78](#)  
DisplayAsIcon, [79](#)

**E**

EmpID, [78-80](#)  
EmpName, [78](#)  
Empty, [79, 80](#)  
EntireRow, [79, 80](#)  
Explicit, [78](#)

**F**

FileCopy, [78](#)  
FileDialog, [78](#)  
FileFldr, [78](#)  
FileName, [78, 79](#)  
FilePath, [79](#)  
FileType, [78, 79](#)  
Filters, [78](#)  
FiltRow, [80, 81](#)  
FollowHyperlink, [80](#)

**H**

Height, [79](#)  
Hidden, [80](#)

**I**

IncrementLeft, [79](#)  
IncrementTop, [79](#)  
Insert, [79](#)  
InStr, [78](#)

**L**

LastAttRow, [78, 80](#)  
LastFiltRow, [80](#)  
LCase, [79](#)  
Left, [79](#)  
Len, [78](#)  
Link, [79](#)  
LockAspectRatio, [79](#)

**M**

MkDir, [78](#)  
MsgBox, [78-80](#)  
msoCTrue, [79](#)  
msoFileDialogFilePicker, [78](#)

**N**

Name, [79](#)  
NoAttach, [79, 80](#)  
NoData, [80, 81](#)  
NoSelection, [78](#)  
Now, [78](#)

**O**

OLEObjects, [79](#)  
OrigFilePath, [78](#)

**P**

Pictures, [79](#)

**R**

Range, [78-81](#)  
Replace, [78](#)  
Right, [78](#)  
Row, [78-80](#)

**S**

SelectedItems, [78](#)  
Shapes, [79](#)  
Sheet1, [78-81](#)  
Sheet3, [78-81](#)  
Sheet4, [78](#)  
Show, [78](#)  
ShowAllData, [80](#)

**T**

ThisWorkbook, [80](#)  
ThumbType, [79](#)  
Title, [78](#)  
Top, [79](#)

**U**

Unique, [80](#)  
UserName, [78](#)

**V**

Value, [78-81](#)  
vbCrLf, [78](#)  
vbDirectory, [78](#)  
Visible, [79](#)

**W**

Width, [79](#)

**X**

xlFilterCopy, [80](#)  
xlFilterInPlace, [80](#)  
xlUp, [78, 80](#)

```

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, 83, 84  
ActiveCell, 83  
ActiveSheet, 84  
ActiveWindow, 83, 84  
ActRow, 83  
Address, 83  
AddTextbox, 83  
Alignment, 83  
AutoShapeType, 84

**C**

Cells, 83, 84  
Collate, 84  
Column, 83  
Copies, 84

**D**

Delete, 84

**E**

EmpIDCard\_CreateTextBox, 83  
EmpIDCard\_Customize, 83  
EmpIDCard\_FieldChange, 83  
EmpIDCard\_Hide, 83  
EmpIDCard\_HideCustom, 83  
EmpIDCard\_HidePic, 83, 84  
EmpIDCard\_Print, 84  
EmpIDCard\_Show, 84  
EmpIDCard\_ShowPic, 83, 84  
EntireColumn, 83, 84  
Explicit, 83

**F**

Fill, 83  
Formula, 83

**H**

Height, 84  
Hidden, 83, 84

**I**

IgnorePrintAreas, 84  
IncrementLeft, 84  
IncrementTop, 84  
Insert, 84

**L**

Left, 84  
Line, 83  
LockAspectRatio, 84

**M**

msoAlignCenter, 83  
msoCTrue, 83, 84  
msoFalse, 83, 84  
msoShapeRoundedRectangle, 84  
msoTextOrientationHorizontal, 83  
msoTrue, 84

**N**

Name, 83, 84  
NoPic, 84

**P**

PageSetup, 84  
ParagraphFormat, 83  
PicPath, 84  
Pictures, 84

PrintArea, 84

PrintOut, 84

**R**

Range, 83, 84  
Row, 83

**S**

ScrollColumn, 83, 84  
Selection, 83  
Shape, 83  
ShapeRange, 84  
Shapes, 83, 84  
Sheet1, 83, 84

**T**

text, 83  
TextBox, 83  
TextBoxName, 83  
TextFrame2, 83  
TextRange, 83  
Top, 84

**V**

Value, 83, 84  
Visible, 83, 84



```

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              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, [86](#), [88](#)  
AdvancedFilter, [86-88](#)  
Application, [86](#), [88](#)  
Apply, [86](#), [88](#)  
Attach\_Refresh, [87](#)

**C**

Cells, [87](#), [89](#)  
Clear, [86](#), [88](#)  
ClearContents, [86-88](#)  
Copy, [86](#), [88](#)  
CopytoRange, [86-88](#)  
CriteriaRange, [86-88](#)  
CutCopyMode, [86](#), [88](#)

**D**

DataOption, [86](#), [88](#)  
Delete, [87](#)

**E**

EmpCol, [86](#), [87](#)  
EmpID, [88](#), [89](#)  
Emp\_EventsRefresh, [86](#), [87](#)  
Emp\_LeaveRefresh, [86](#), [87](#)  
Emp\_Load, [86](#)  
Emp\_PayItemsRefresh, [87](#)  
Emp\_TimeclockRefresh, [87](#), [88](#)  
Emp\_WeeklySchedRefresh, [87](#), [88](#)  
EmpMgrRow, [87](#), [88](#)  
EmpRow, [86](#), [87](#)  
Empty, [86](#), [88](#)  
Explicit, [86](#)

**F**

Find, [89](#)  
FoundWeeklySchedID, [88](#), [89](#)

**I**

ItemType, [87](#)

**K**

Key, [86](#), [88](#)

**L**

LastEventRow, [86](#)  
LastFiltRow, [86-88](#)  
LastLeaveRow, [86](#)  
LastPayItemRow, [87](#)  
LastTimeclockRow, [88](#)

**M**

MsgBox, [87](#)  
msoCTrue, [87](#)  
msoFalse, [87](#)

**N**

NoEvents, [86](#)  
NoLeave, [86](#)  
NoPayItem, [87](#), [88](#)  
NoTimeclock, [88](#)  
NoWeekly, [89](#)

**O**

Order, [86](#), [88](#)

**P**

PasteSpecial, [86](#), [88](#)  
PayItemRow, [87](#), [88](#)

**R**

Range, [86-89](#)  
ResetCalc, [87](#), [88](#)  
Row, [86-89](#)

**S**

SchedRow, [88](#), [89](#)  
SetRange, [86](#), [88](#)  
Shapes, [87](#)  
Sheet1, [86-88](#)  
Sheet11, [86](#)  
Sheet13, [88](#), [89](#)  
Sheet17, [87](#)  
Sheet2, [87](#)  
Sheet7, [86](#)  
Sheet9, [88](#)  
Show\_EmplPic, [87](#)  
Sort, [86](#), [88](#)  
SortFields, [86](#), [88](#)  
SortOn, [86](#), [88](#)  
StopCalc, [87](#), [88](#)

**U**

Unique, [86-88](#)

**V**

Value, [86-89](#)  
Visible, [87](#)

**W**

WeeklyCol, [88](#), [89](#)  
WeeklyRow, [88](#), [89](#)  
WeeklySchedID, [88](#), [89](#)  
WeeklySchedRow, [88](#)

**X**

xlCopy, [86](#), [88](#)  
xlDescending, [86](#), [88](#)  
xlFilterCopy, [86-88](#)  
xlPasteFormulas, [86](#), [88](#)  
xlSortNormal, [86](#), [88](#)  
xlSortOnValues, [86](#), [88](#)  
xlUp, [86-88](#)  
xlValues, [89](#)  
xlWhole, [89](#)

Option Explicit

Sub Empl\_Archive()

Dim EmplRow As Long

Dim ArchiveRow As Long

Dim EmplName As String

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

With Sheet1

If .Range("B4" ).Value = Empty Then

MsgBox "Please select an employee to Archive"

Exit Sub

End If

EmplName = .Range("F2" ).Value 'Employee Name

EmplRow = .Range("B4" ).Value 'Employee Row

ArchiveRow = Sheet6.Range("A99999" ).End(xlUp).Row + 1 'First Available Archive Row

.Range("H14" ).Value = "Inactive" 'You can change this to whatever you want

Sheet6.Range("A" & ArchiveRow & ":AG" & ArchiveRow).Value = Sheet2.Range("A" &

EmplRow & ":AG" & EmplRow).Value 'Copy Over Employee Data

Sheet2.Range(EmplRow & ":" & EmplRow).EntireRow.Delete

Sheet1.Range("F2" ).Value = Sheet2.Range("BC4" ).Value 'Reset to display First Employee

'Update Employee List in the Schedule

Sheet8.Range("D8" ).Formula = "='Employee List'!BC4"

Sheet8.Range("D8:D314" ).Formula = Sheet8.Range("D8" ).Formula 'Update Employee

Schedule Names

MsgBox EmplName & " has been archived"

End With

End Sub

Sub Empl\_CancelNew()

Sheet1.Range("F2" ).Value = Sheet2.Range("BC4" ).Value 'Reset to display First Employee

End Sub

Sub Empl\_New()

With Sheet1

.Range("B1" ).Value = True 'Set Employee Load To False

.Shapes("NewEmpGrp" ).Visible = msoCTrue

.Shapes("ExistEmpGrp" ).Visible = msoFalse

.Shapes("DefaultPicture" ).Visible = msoCTrue

On Error Resume Next

.Shapes("EmplPic" ).Delete

On Error GoTo 0

.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

.Range(

"F26:G26, I26, K26, E19:K24, E28:L44, F46:G46, I46, K46, E48:L64, G93: I104, K93:K104, G108: I119, K108:K119, G128: I139, K128:K139, F148: J154, L148: L154, E168: H179, L168: HL179, E188: K204" ).

ClearContents

'Add New Employee Default Values

.Range("J2" ).Value = Application.WorksheetFunction.Max(Sheet2.Range("EmployeeID" )) +

1

.Range("B6" ).Value = True 'Set New Employee To True

.Range("E4" ).Select 'Selects General Info Tab

.Range("F6" ).Select 'Select Last Name Field

```

1 2
50     .Range("B1" ).Value = False 'Set Employee Load To False
51
52     End With
53 End Sub
54
55 Sub Empl_ReActivate()
56     Dim EmplRow As Long
57     Dim ArchiveRow As Long
58     Dim EmplName As String
59     If MsgBox("Are you sure you want to reactivate this employee?" , vbYesNo, "Reactivate
Employee" ) = vbNo Then Exit Sub
60     With Sheet6
61         If .Range("BA1" ).Value = Empty Then
62             MsgBox "Please select an employee to Reactivate"
63             Exit Sub
64         End If
65
66         ArchiveRow = .Range("BA1" ).Value 'Employee Row
67         EmplRow = Sheet6.Range("A99999" ).End(xlUp).Row + 1 'First Available Archive Row
68         .Range("J" & ArchiveRow).Value = "Active" 'You can change this to whatever you want
69         Sheet2.Range("A" & EmplRow & ":AG" & EmplRow).Value = .Range("A" & ArchiveRow &
":AG" & ArchiveRow).Value 'Copy Over From Archive List
70         EmplName = Sheet2.Range("BC" & EmplRow).Value 'Employee Name
71         .Range(ArchiveRow & ":" & ArchiveRow).EntireRow.Delete 'Delete Old Archive Row
72         'Update Employee List in the Schedule
73         Sheet8.Range("D8" ).Formula = "'Employee List'!BC4"
74         Sheet8.Range("D8:D314" ).Formula = Sheet8.Range("D8" ).Formula 'Update Employee
Schedule Names
75         Empl_SortByName
76         MsgBox EmplName & " has been reactivated"
77     End With
78 End Sub
79
80
81 Sub Empl_SaveNew()
82     Dim EmpRow As Long
83     Dim EmpCol As Long
84     With Sheet1
85         If .Range("F6" ).Value = Empty Then
86             MsgBox "Please enter a Last Name for the Employee"
87             .Range("F6" ).Select
88             Exit Sub
89         End If
90         EmpRow = Sheet2.Range("A99999" ).End(xlUp).Row + 1 'First Avail Row
91         Sheet2.Range("A" & EmpRow).Value = Application.WorksheetFunction.Max(Sheet2.Range(
"EmployeeID" )) + 1
92         For EmpCol = 2 To 33
93             Sheet2.Cells(EmpRow, EmpCol).Value = .Range(Sheet2.Cells(1, EmpCol).Value).Value
94         Next EmpCol
95         .Range("F2" ).Value = .Range("F6" ).Value & ", " & .Range("I6" ).Value 'Add in Last
Name, First Name
96
97         .Shapes("NewEmpGrp" ).Visible = msoFalse
98         .Shapes("ExistEmpGrp" ).Visible = msoCTrue
99         .Range("B1" ).Value = False 'Set Empl Load To False
100        .Range("B6" ).Value = False 'Set New Employee To False
101    End With
102    Empl_SortByName 'Sort Employee List
103    Sheet8.Range("D8:D314" ).Formula = Sheet8.Range("D8" ).Formula 'Update Employee Schedule
Names
104 End Sub

```

```
105
106
107
108 Sub Empl_SortByName()
109     'Sort Employee List
110     StopCalc
111     Dim EmpRow As Long
112     With Sheet2
113         EmpRow = .Range("A99999").End(xlUp).Row 'Last Row of value
114         .Sort.SortFields.Clear
115         .Sort.SortFields.Add Key:=.Range("B4"), SortOn:=xlSortOnValues, Order:=xlAscending,
116         DataOption:=xlSortNormal
117         With .Sort
118             .SetRange Range("A4:AG" & EmpRow)
119             .Apply
120         End With
121     End With
122     ResetCalc
123 End Sub
```



**A**

Add, 93  
Application, 91, 92  
Apply, 93  
ArchiveRow, 91, 92

**C**

Cells, 92  
Clear, 93  
ClearContents, 91

**D**

DataOption, 93  
Delete, 91, 92

**E**

EmpCol, 92  
Empl\_Archive, 91  
Empl\_CancelNew, 91  
Empl\_New, 91  
Empl\_ReActivate, 92  
Empl\_SaveNew, 92  
Empl\_SortByName, 92, 93  
EmplName, 91, 92  
EmplRow, 91, 92  
EmpRow, 92, 93  
Empty, 91, 92  
EntireRow, 91, 92  
Explicit, 91

**F**

Formula, 91, 92

**K**

Key, 93

**M**

Max, 91, 92  
MsgBox, 91, 92  
msoCTrue, 91, 92  
msoFalse, 91, 92

**O**

Order, 93

**R**

Range, 91-93  
ResetCalc, 93  
Row, 91-93

**S**

SetRange, 93  
Shapes, 91, 92  
Sheet1, 91, 92  
Sheet2, 91-93  
Sheet6, 91, 92  
Sheet8, 91, 92  
Sort, 93  
SortFields, 93  
SortOn, 93  
StopCalc, 93

**V**

Value, 91, 92  
vbCrLf, 91  
vbNo, 91, 92  
vbYesNo, 91, 92  
Visible, 91, 92

**W**

WorksheetFunction, 91, 92

**X**

xlAscending, 93  
xlSortNormal, 93  
xlSortOnValues, 93  
xlUp, 91-93

```

1  Option Explicit
2
3  Sub Add_EmplPic()
4      Dim PicFile As FileDialog
5      Dim FileName, OrigFilePath, DestFileFolder, DestFilePath, EmpName As String
6      With Sheet1
7          EmpName = Replace(Replace(.Range("F2").Value, ",", "_"), " ", "" ) 'Get Clean
            Employee Name without commas or spaces
8          Set PicFile = Application.FileDialog(msoFileDialogFilePicker) 'Open File Dialog
9          With PicFile
10             .Title = "Select an Employee Picture"
11             .Filters.Add "All Picture Files", "*.jpg,*jpeg,*gif*.png,*gif,*bmp*.tiff"
12             If .Show <> -1 Then GoTo NoSelection
13             OrigFilePath = .SelectedItems(1)
14         End With
15         DestFileFolder = Sheet4.Range("EmplAttachFolder").Value 'Get Main Folder for Employee
            pictuers and attachments
16         If DestFileFolder = "" Then 'If user has not added a main folder yet
17             MsgBox "Before adding an Employee Picture, please select a dedicated" & vbCrLf &
                "folder to store the individual Employee Attachments"
18             Exit Sub
19         End If
20         FileName = Dir(OrigFilePath)
21         'Check if Employee Folder exists, if not, add it.
22         If Len(Dir(DestFileFolder & "\ " & EmpName, vbDirectory)) = 0 Then
23             MkDir DestFileFolder & "\ " & EmpName 'Create Directory if it does not exist
24         End If
25         DestFilePath = DestFileFolder & "\ " & EmpName & "\ " & FileName
26         FileCopy OrigFilePath, DestFilePath 'Copy user selected file into appropriate
            Employee folder
27         'Kill (OrigFilePath) 'If you want to delete the original file in the original location
28         Sheet1.Range("K10").Value = DestFilePath 'Put File Name in K10
29         Show_EmplPic
30         NoSelection:
31     End With
32 End Sub
33
34
35 Sub Show_EmplPic()
36     Dim PicPath As String
37     With Sheet1
38         On Error Resume Next
39         .Shapes("EmplPic").Delete 'Delete Picture if it exists
40         On Error GoTo 0
41         PicPath = .Range("K10").Value 'Path of the picture
42         If PicPath = Empty Then
43             .Shapes("DefaultPicture").Visible = msoCTrue
44             Exit Sub
45         End If
46         .Shapes("DefaultPicture").Visible = msoFalse
47         On Error GoTo DefaultPic
48         With .Pictures.Insert(PicPath)
49             With .ShapeRange
50                 .LockAspectRatio = msoTrue
51                 .Height = 95
52                 .Name = "EmplPic"
53             End With 'Shape Range
54         End With 'Pictures
55
56     With .Shapes("EmplPic")

```

1 2 3

```
57     1 2 3 .Left = Sheet1.Range("J5" ).Left
58     .Top = Sheet1.Range("J5" ).Top
59     .IncrementLeft 40
60     .IncrementTop 10
61     End With
62     Exit Sub:
63     DefaultPic:
64     .Shapes("DefaultPicture" ).Visible = msoCTrue
65     End With 'Sheet 1
66
67     End Sub
```

## A

Add, [95](#)  
Add\_EmplPic, [95](#)  
Application, [95](#)

## D

DefaultPic, [95](#), [96](#)  
Delete, [95](#)  
DestFileFolder, [95](#)  
DestFilePath, [95](#)  
Dir, [95](#)

## E

EmpName, [95](#)  
Empty, [95](#)  
Explicit, [95](#)

## F

FileCopy, [95](#)  
FileDialog, [95](#)  
FileName, [95](#)  
Filters, [95](#)

## H

Height, [95](#)

## I

IncrementLeft, [96](#)  
IncrementTop, [96](#)  
Insert, [95](#)

## L

Left, [96](#)  
Len, [95](#)  
LockAspectRatio, [95](#)

## M

MkDir, [95](#)  
MsgBox, [95](#)  
msoCTrue, [95](#), [96](#)  
msoFalse, [95](#)  
msoFileDialogFilePicker, [95](#)  
msoTrue, [95](#)

## N

Name, [95](#)  
NoSelection, [95](#)

## O

OrigFilePath, [95](#)

## P

PicFile, [95](#)  
PicPath, [95](#)  
Pictures, [95](#)

## R

Range, [95](#), [96](#)  
Replace, [95](#)

## S

SelectedItems, [95](#)  
ShapeRange, [95](#)  
Shapes, [95](#), [96](#)  
Sheet1, [95](#), [96](#)  
Sheet4, [95](#)  
Show, [95](#)  
Show\_EmplPic, [95](#)

## T

Title, [95](#)  
Top, [96](#)

## V

Value, [95](#)  
vbCrLf, [95](#)  
vbDirectory, [95](#)  
Visible, [95](#), [96](#)

```

1 Option Explicit
2
3 Sub Empl_SwitchHorizontalTabs()
4     Dim SelCol As Long
5     Dim FirstRow As Long
6     StopCalc
7     With Sheet1
8         SelCol = .Range("B2" ).Value
9
10        If SelCol > 8 Then SelCol = SelCol + 3
11        .Range("5:204" ).EntireRow.Hidden = True
12        FirstRow = 5 + ((SelCol - 5) * 20)
13        .Range(FirstRow & ":" & FirstRow + 19).EntireRow.Hidden = False
14        If SelCol <> 5 Then
15            On Error Resume Next
16            .Shapes("GenInfoGrp" ).Visible = msoFalse
17            On Error GoTo 0
18            .Shapes("DefaultPicture" ).Visible = msoFalse
19            On Error Resume Next
20            .Shapes("EmplPic" ).Visible = msoFalse
21            On Error GoTo 0
22            On Error Resume Next
23            .Shapes("AttachThumb" ).Visible = msoFalse
24            On Error GoTo 0
25        Else:
26            On Error Resume Next
27            .Shapes("GenInfoGrp" ).Visible = msoCTrue
28            On Error GoTo 0
29            If .Range("K10" ).Value = Empty Then .Shapes("DefaultPicture" ).Visible = msoCTrue
30            Else: Show_EmplPic
31        End If
32        If SelCol = 8 Then .Range("B3" ).Value = 66
33        'Select First Cell in Tab
34        If .Range("B2" ).Value = 5 Then .Range("F6" ).Select
35        If .Range("B2" ).Value = 6 Then .Range("F26" ).Select
36        If .Range("B2" ).Value = 7 Then .Range("F46" ).Select
37        If .Range("B2" ).Value = 8 Then .Range("G66" ).Select
38        If .Range("B2" ).Value = 9 Then .Range("F148" ).Select
39        If .Range("B2" ).Value = 10 Then .Range("E168" ).Select
40        If .Range("B2" ).Value = 11 Then .Range("E188" ).Select
41    End With
42    ResetCalc
43 End Sub
44 Sub EmpTab_Events()
45     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("K4" ).Select
46 End Sub
47
48 Sub EmpTab_GenInfo()
49     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("E4" ).Select
50 End Sub
51 Sub EmpTab_Leave()
52     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("J4" ).Select
53 End Sub
54 Sub EmpTab_PayDetails()
55     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("H4" ).Select
56 End Sub
57 Sub EmpTab_PayHistory()
58     If ActiveSheet.CodeName = "Sheet1" Then Sheet1.Range("G4" ).Select
59 End Sub
60 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, [99](#)  
ActiveSheet, [98](#), [99](#)

## C

CodeName, [98](#), [99](#)

## E

Empl\_SwitchHorizontalTabs, [98](#)  
EmpTab\_Events, [98](#)  
EmpTab\_GenInfo, [98](#)  
EmpTab\_Leave, [98](#)  
EmpTab\_PayDetails, [98](#)  
EmpTab\_PayHistory, [98](#)  
EmpTab\_Scheduling, [98](#)  
EmpTab\_TimeClock, [99](#)  
Empty, [98](#)  
EntireRow, [98](#), [99](#)  
Explicit, [98](#)

## F

FirstRow, [98](#), [99](#)

## H

Hidden, [98](#), [99](#)

## M

msoCTrue, [98](#)  
msoFalse, [98](#)

## R

Range, [98](#), [99](#)  
ResetCalc, [98](#)  
Right, [99](#)  
Row, [99](#)

## S

SelCol, [98](#)  
SelRow, [99](#)  
Shapes, [98](#)  
Sheet1, [98](#), [99](#)  
Show\_EmplPic, [98](#)  
StopCalc, [98](#)  
SwitchVerticalTabs, [99](#)

## V

Value, [98](#), [99](#)  
Visible, [98](#)

```
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, [101](#)

**E**

Event\_Refresh, [101](#)

Events\_ClearFilter, [101](#)

Events\_FilterTable, [101](#)

Explicit, [101](#)

**I**

InStr, [101](#)

**R**

Range, [101](#)

**S**

Sheet5, [101](#)

Sheet7, [101](#)

**V**

Value, [101](#)

```

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
59 Then Exit Sub
60 With Sheet5
61     If .Range("B3").Value = Empty Then Exit Sub
62     EventRow = .Range("B3").Value 'Event Row
63     Sheet7.Range(EventRow & ":" & EventRow).EntireRow.Delete
64     Event_Refresh 'Refresh Events
65     .Range("E27").Select
66 End With
67 End Sub
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             msoTrue 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 = msoTrue 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

```

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, [103](#), [105](#)  
AdvancedFilter, [105](#)  
Application, [103](#), [105](#)  
Apply, [105](#)

**C**

Cells, [103-105](#)  
Characters, [104](#)  
Clear, [105](#)  
ClearContents, [103](#), [105](#)  
Copy, [105](#)  
CopytoRange, [105](#)  
CriteriaRange, [105](#)  
CutCopyMode, [105](#)

**D**

DataOption, [105](#)  
Delete, [103](#), [104](#)  
DeleteFilters, [103](#), [105](#)

**E**

Empty, [103-105](#)  
EntireRow, [104](#)  
Event\_AddEmailAttach, [103](#)  
Event\_AddNew, [103](#)  
Event\_AddNotifDefaults, [103](#)  
Event\_AllDay, [103](#)  
Event\_CancelNew, [103](#)  
Event\_Delete, [103](#)  
Event\_Load, [104](#)  
Event\_Refresh, [104](#), [105](#)  
Event\_SaveNew, [105](#)  
EventCol, [103-105](#)  
EventRow, [103-105](#)  
Explicit, [103](#)

**F**

FileDialog, [103](#)  
FileFldr, [103](#)  
Filters, [103](#)  
Font, [104](#)  
Formula, [105](#)

**I**

InStr, [105](#)

**K**

Key, [105](#)

**L**

LastEventRow, [104](#), [105](#)  
LastFiltRow, [104](#), [105](#)  
Length, [104](#)

**M**

MapRng, [103](#), [104](#)  
MsgBox, [103-105](#)  
msoCTrue, [103](#), [104](#)  
msoFalse, [103-105](#)  
msoFileDialogFilePicker, [103](#)  
msoTrue, [104](#), [105](#)

**N**

Name, [104](#)  
Names, [103](#)  
NoEvent, [105](#)  
NoSelection, [103](#)  
NotifCol, [103](#)

NotifRow, [103](#)  
Now, [103](#)

**O**

Order, [105](#)

**P**

PasteSpecial, [105](#)

**R**

Range, [103-105](#)  
ResetCalc, [104](#), [105](#)  
Row, [105](#)

**S**

SelectedItems, [103](#)  
SelRow, [104](#)  
SetRange, [105](#)  
Shapes, [103-105](#)  
Sheet4, [103](#)  
Sheet5, [103-105](#)  
Sheet7, [103-105](#)  
Show, [103](#)  
Sort, [105](#)  
SortFields, [105](#)  
SortOn, [105](#)  
Start, [104](#)  
StopCalc, [104](#)

**T**

Title, [103](#)

**U**

Unique, [105](#)

**V**

Value, [103-105](#)  
vbNo, [103](#)  
vbYesNo, [103](#)  
Visible, [103-105](#)

**W**

WrapText, [105](#)

**X**

xlCopy, [105](#)  
xlDescending, [105](#)  
xlFilterCopy, [105](#)  
xlPasteFormulas, [105](#)  
xlSortNormal, [105](#)  
xlSortOnValues, [105](#)  
xlUp, [105](#)

```
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, [107](#)

**D**

delay, [107](#)

DoEvents, [107](#)

**E**

EmployIDNotFound, [107](#)

Explicit, [107](#)

**F**

Fill, [107](#)

**I**

i, [107](#)

**S**

Shapes, [107](#)

StartTime, [107](#)

**T**

Timer, [107](#)

Transparency, [107](#)

**V**

Visible, [107](#)

```
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, [109](#)  
AllowMultiSelect, [109](#)  
Application, [109](#)

## B

BeregnerNm, [109](#)  
BeregnerRng, [109](#)

## E

EmailAdd, [109](#)  
Explicit, [109](#)

## F

FileDialog, [109](#)  
Find, [109](#)  
FoundBeregner, [109](#)

## G

GetFolder, [109](#)

## I

InitialFileName, [109](#)

## M

MailToSelected, [109](#)  
msoFileDialogFolderPicker, [109](#)

## N

NextCode, [109](#)

## P

PDFFldr, [109](#)

## R

Range, [109](#)  
Row, [109](#)

## S

SelectedItems, [109](#)  
SelRow, [109](#)  
Sheet1, [109](#)  
Sheet2, [109](#)  
Show, [109](#)  
sItem, [109](#)  
strPath, [109](#)

## T

Title, [109](#)

## V

Value, [109](#)

## X

xlValues, [109](#)  
xlWhole, [109](#)

---

```
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
28        With .Sort
29            .SetRange Range("CA5:CW" & LastFiltRow)
30            .Apply
31        End With
32        If .Range("CV5").Value < Now Then 'Send Now
33            Notifications_Create
34        Else: 'Schedule Send
35            ' Application.OnTime .Range("CV5").Value, "Notifications_Create"
36        End If
37        NoNotif:
38    End With
39 End Sub
```

---

```
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(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(
NotifText, "[Event Name]" , EventName), "[Event Type]" , EventType), "[Event
Notes]" , EventNotes), "[Event Created On Date]" , EventCreatedOn), "[Start/Next
Date & Time]" , StartOn), _
114     "[End Date & Time]" , EndOn), "[Employee Name]" , EmployeeName), "[User Name]" ,
UserName), "[Leave Type]" , LeaveType)
115     .Range("CY5" ).Value = NotifText
116     .Range(5 & ":" & 5).WrapText = False
117     Sheet12.Shapes("NotificationGrp" ).Copy
118     ActiveSheet.Range("A1" ).Select
119     ActiveSheet.Paste
120     With ActiveSheet.Shapes("NotificationGrp" )
121         .Height = 172
122         .Width = 197
123         .Left = ActiveCell.Left
124         .Top = ActiveCell.Top
125         .Visible = msoCTrue
126     End With
127     ActiveSheet.Shapes("YellowNote" ).TextFrame.Characters.text = NotifText
128     ActiveSheet.Range("A1" ).Select
129 End If
130
131 'Set Reminded On Date & time
132 .Range("R" & EventRow).Value = Now 'Set Reminded On Date & Time To Current Date &
Time
133
134 End With
135 Notifications_Check 'Recheck List
136 End Sub
137
138 Sub testting()
139     With ActiveSheet.Shapes("NotificationGrp" )
140         MsgBox "Height:" & .Height & vbCrLf & "Width:" & .Width
141     End With
142 End Sub
143 End Sub
144
145
146
```

—, 112, 113

## A

ActiveCell, 113  
 ActiveSheet, 111, 113  
 Add, 111, 112  
 AdvancedFilter, 111  
 Apply, 111  
 Attach, 111, 112  
 Attachments, 112

## B

BCC, 112  
 Body, 112

## C

Characters, 113  
 Clear, 111  
 ClearContents, 111  
 Copy, 113  
 CopytoRange, 111  
 CreateItem, 112  
 CreateObject, 112  
 CriteriaRange, 111

## D

DataOption, 111  
 Delete, 111  
 Display, 112

## E

EmailBCC, 111, 112  
 EmailTo, 111, 112  
 EmplID, 111, 112  
 EmplIDRng, 111, 112  
 EmployeeEmail, 111, 112  
 EmployeeName, 111-113  
 EndOn, 111-113  
 EventCreatedOn, 111-113  
 EventName, 111-113  
 EventNotes, 111-113  
 EventRow, 111-113  
 EventType, 111-113  
 Explicit, 111

## F

Find, 112  
 FoundEmplRng, 112

## H

Height, 113

## K

Key, 111

## L

LastFiltRow, 111  
 LastRow, 111  
 LeaveType, 111-113  
 Left, 113

## M

Mess, 111, 112  
 MsgBox, 113  
 msoCTrue, 113

## N

NoNotif, 111

Notification\_Close, 111  
 Notifications\_Check, 111, 113  
 Notifications\_Create, 111  
 NotifText, 111-113  
 Now, 111, 113

## O

Order, 111  
 OutApp, 111, 112  
 OutMail, 111-113

## P

Paste, 113

## R

Range, 111-113  
 Replace, 112, 113  
 Row, 111, 112

## S

SetRange, 111  
 Shapes, 111, 113  
 Sheet12, 113  
 Sheet14, 112  
 Sheet2, 112  
 Sheet7, 111, 112  
 Sort, 111  
 SortFields, 111  
 SortOn, 111  
 StartOn, 111-113  
 Subj, 111, 112  
 Subject, 112

## T

testting, 113  
 text, 113  
 TextFrame, 113  
 Top, 113

## U

Unique, 111  
 UserEmail, 111, 112  
 UserName, 111-113

## V

Value, 111-113  
 vbCrLf, 113  
 Visible, 113

## W

Width, 113  
 WrapText, 113

## X

xlAscending, 111  
 xlFilterCopy, 111  
 xlSortNormal, 111  
 xlSortOnValues, 111  
 xlUp, 111  
 xlValues, 112  
 xlWhole, 112

Option Explicit

Sub Payroll\_Load\_Review()

Dim PayrollRow As Long

Dim LastPayEmpRow As Long

Dim LastFiltRow

With Sheet16

If .Range("B15").Value = Empty Then Exit Sub ' Exit On No Payroll Row Found

.Range("B1").Value = True 'Set Payroll Load to True

.Range("B2").Value = .Range("B15").Value 'Set Payroll ID

StopCalc

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

.Range("AJ8,AE14:AH44,AK9:AK44,AL14:AL25,AP14:AQ25,AL29:AM40,AP29:AQ40").ClearContents  
'clear fields

.Range("AA9:AD310,AO14:A040,AS14:AS25,AW29:AW40").ClearContents 'clear fields

.Range("AM3").Value = Sheet18.Range("C" & PayrollRow).Value 'Payroll Date

.Range("J6").Value = Sheet18.Range("D" & PayrollRow).Value 'From Date

.Range("J8").Value = Sheet18.Range("E" & PayrollRow).Value 'To Date

.Range("AN3").Value = Sheet18.Range("F" & PayrollRow).Value 'Payroll Status

'Add Employees

Sheet20.Range("L3:Q9999").ClearContents 'Clear Prev Criteria & filters

Sheet20.Range("L3").Value = .Range("B2").Value 'Payroll ID

LastPayEmpRow = Sheet20.Range("A9999").End(xlUp).Row 'Last Employee Row

Sheet20.Range("A2:E" & LastPayEmpRow).AdvancedFilter xlFilterCopy, CriteriaRange:=  
Sheet20.Range("L2:L3"), CopytoRange:=Sheet20.Range("M2:Q2"), Unique:=True

LastFiltRow = Sheet20.Range("M9999").End(xlUp).Row

If LastFiltRow < 3 Then GoTo NoData

.Range("AA9:AB" & LastFiltRow + 6).Value = Sheet20.Range("M3:N" & LastFiltRow).Value

'Add Status & Employee ID

.Range("AD9:AD" & LastFiltRow + 6).Value = Sheet20.Range("Q3:Q" & LastFiltRow).Value

'Add Table Rows

.Range("AC7").Copy

.Range("AC9:AC" & LastFiltRow + 6).PasteSpecial xlPasteFormulas

.Calculate

.Range("AC9:AC" & LastFiltRow + 6).Copy

.Range("AC9").PasteSpecial xlPasteValues

Application.CutCopyMode = xlCopy

NoData:

.Range("B1").Value = False 'Set Payroll Load to False

ResetCalc

.Range("AC9").Select

End With

End Sub

Sub Payroll\_Load\_Summary()

Dim PayrollRow As Long

Dim LastPayEmpRow As Long

Dim LastFiltRow

With Sheet16

If .Range("B15").Value = Empty Then Exit Sub ' Exit On No Payroll Row Found

.Range("B1").Value = True 'Set Payroll Load to True

.Range("B2").Value = .Range("B16").Value 'Set Payroll ID

StopCalc

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

.Range("BC9:BC999,BD10:BI999").ClearContents 'Clear fields

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

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

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

.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, [115](#), [116](#)  
Application, [115](#)

## C

Calculate, [115](#)  
ClearContents, [115](#), [116](#)  
Copy, [115](#)  
CopytoRange, [115](#), [116](#)  
CriteriaRange, [115](#), [116](#)  
CutCopyMode, [115](#)

## E

Empty, [115](#)  
Explicit, [115](#)

## F

Formula, [116](#)

## L

LastFilterRow, [115](#), [116](#)  
LastPayEmpRow, [115](#), [116](#)

## N

NoData, [115](#), [116](#)

## P

PasteSpecial, [115](#)  
Payroll\_Load\_Review, [115](#)  
Payroll\_Load\_Summary, [115](#)  
PayrollRow, [115](#)

## R

Range, [115](#), [116](#)  
ResetCalc, [115](#), [116](#)  
Row, [115](#), [116](#)

## S

Sheet16, [115](#)  
Sheet18, [115](#)  
Sheet20, [115](#), [116](#)  
StopCalc, [115](#)

## U

Unique, [115](#), [116](#)

## V

Value, [115](#), [116](#)

## X

xlCopy, [115](#)  
xlFilterCopy, [115](#), [116](#)  
xlPasteFormulas, [115](#)  
xlPasteValues, [115](#)  
xlUp, [115](#), [116](#)



```

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("AO14: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
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, 121  
 Add, 118  
 AdvancedFilter, 118, 119  
 Apply, 118

**B**

BreakIn, 121  
 BreakInVal, 120, 121  
 BreakOut, 121  
 BreakOutVal, 120, 121

**C**

Clear, 118  
 ClearContents, 118, 119  
 ClockIn, 121  
 ClockInVal, 120, 121  
 ClockOut, 121  
 ClockOutVal, 120, 121  
 CopytoRange, 118, 119  
 CriteriaRange, 118, 119

**D**

DataOption, 118  
 DateRow, 120

**E**

EditTimeClock, 121  
 EditTimeClock\_Calculate, 121  
 EmpID, 118, 119, 121  
 EmpPayRow, 118, 119  
 EmpRow, 118  
 Empty, 121  
 Explicit, 118

**I**

Int, 120  
 ItemType, 118-120

**K**

Key, 118

**L**

LastFiltRow, 118, 119  
 LastLeaveRow, 119  
 LastPayItemRow, 118, 119  
 LastTimeclockRow, 118  
 Left, 121

**M**

msoCTrue, 121

**N**

NoLeave, 119  
 NoPayData, 119, 120  
 NoPayItem, 119  
 Notes, 121  
 NoTimeclock, 118

**O**

Order, 118

**P**

PayItemRow, 118-120  
 Payroll\_LoadEmpReview, 118  
 Payroll\_Review\_EditTimeClockEntry, 120  
 Payroll\_ShowEditEmployeeIcon, 121  
 Payroll\_ShowEditTimeClockIcon, 121  
 Payroll\_ViewEmployeeInfo, 121

PayrollID, 118, 119  
 PayrollRow, 118-120

**R**

Range, 118-121  
 ResetCalc, 120  
 ReviewRow, 118  
 Row, 118, 119

**S**

SelRow, 120, 121  
 SetRange, 118  
 Shapes, 121  
 Sheet1, 121  
 Sheet11, 119  
 Sheet16, 118, 120, 121  
 Sheet17, 119  
 Sheet19, 119, 120  
 Sheet2, 118  
 Sheet8, 120  
 Sheet9, 118, 120, 121  
 Show, 121  
 Sort, 118  
 SortFields, 118  
 SortOn, 118  
 StopCalc, 118

**T**

TCRow, 120, 121  
 Top, 121

**U**

Unique, 118, 119

**V**

Value, 118-121  
 Visible, 121

**X**

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

```
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
157   For PayrollRow = 29 To 40
158     If .Range("AL" & PayrollRow).Value = Empty Then Exit For
159     If .Range("AO" & PayrollRow).Value = Empty Then
160       PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Pay
161       Item Row
162       Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
163       Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
164       Sheet19.Range("C" & PayItemRow).Value = "Other Addition" 'Set Other Addition
165       Item Type
166       Sheet19.Range("G" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
167         PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
168       locate Payroll Date
169       Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
170       Sheet19.Range("I" & PayItemRow).Value = PayrollRow
171       Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
172       .Range("AO" & PayrollRow).Value = PayItemRow
173     Else: 'Set Existing Pay Item Row
174       PayItemRow = .Range("AO" & PayrollRow).Value
175     End If
176     Sheet19.Range("D" & PayItemRow).Value = .Range("AL" & PayrollRow).Value 'Other
177     Deduction Name
178     Sheet19.Range("E" & PayItemRow).Value = .Range("AM" & PayrollRow).Value 'Other
179     Deduction Amount
180   Next PayrollRow
181
182   'Add In Employee Paid Leave Items
183   For PayrollRow = 29 To 40
184     If .Range("AP" & PayrollRow).Value = Empty Then Exit For
185     If .Range("AQ" & PayrollRow).Value <> Empty Then 'Only add items when they contain
186     hours
187     If .Range("AW" & PayrollRow).Value = Empty Then
188       PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Pay
189       Item Row
190       Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
191       Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
192       Sheet19.Range("C" & PayItemRow).Value = "Paid Leave" 'Set Other Addition
193       Item Type
194       Sheet19.Range("G" & PayItemRow).Value =
195         "=IFERROR(INDEX(Payroll_Date,MATCH(A" & PayItemRow & ",Payroll_ID,0),1)," &
196         Chr(34) & Chr(34) & ")" 'Use formula to locate Payroll Date
197       Sheet19.Range("H" & PayItemRow).Value = "=Year(G" & PayItemRow & ")"
198       Sheet19.Range("I" & PayItemRow).Value = PayrollRow
199       Sheet19.Range("J" & PayItemRow).Value = "=Row()" 'Set Table Row
200       .Range("AU" & PayrollRow).Value = PayItemRow
201     Else: 'Set Existing Pay Item Row
202       PayItemRow = .Range("AW" & PayrollRow).Value
203     End If
204     Sheet19.Range("D" & PayItemRow).Value = .Range("AP" & PayrollRow).Value 'Paid
205     Leave Name
206     Sheet19.Range("F" & PayItemRow).Value = .Range("AQ" & PayrollRow).Value 'Paid
207     Leave Hours
208   End If 'Check for Hours To Use in AQ
209   Next PayrollRow
210   .Range("AA" & ReviewRow).Value = "ü" 'Add Completed Checkbox
211   If .Range("AD" & ReviewRow).Value <> Empty Then Sheet20.Range("C" & .Range("AD" &
212   ReviewRow).Value).Value = "ü" 'Update Status in Employee Payroll List
213   ResetCalc
214
215   If .Range("B19").Value = 1 Then 'Update Status on All Employees Reviewed
216     .Range("AN3").Value = "Pending Confirmation"
217     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, [123](#)

## C

Chr, [124-126](#)

## E

EmpID, [123-126](#)

Empty, [123-127](#)

Explicit, [123](#)

## L

LastPayrollEmpRow, [123](#)

## M

MsgBox, [123](#), [127](#)

## P

PayEmpListRow, [123](#)

PayItemRow, [123-126](#)

Payroll\_PrevEmployee, [123](#)

Payroll\_SaveEmployeePay, [123](#)

PayrollEmpRow, [123](#), [124](#)

PayrollID, [123-126](#)

PayrollListRow, [123](#), [124](#), [126](#)

PayrollRow, [123](#), [125](#), [126](#)

PayType, [123](#), [124](#)

## R

Range, [123-127](#)

ResetCalc, [126](#)

ReviewRow, [123](#), [124](#), [126](#), [127](#)

Row, [123-126](#)

## S

Sheet16, [123](#)

Sheet18, [123](#), [126](#)

Sheet19, [124-126](#)

Sheet20, [123](#), [126](#)

StopCalc, [123](#)

## T

text, [123](#)

## V

Value, [123-127](#)

## W

WorksheetFunction, [123](#)

## X

xlUp, [123-126](#)

```

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

```

```

60 1 2 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
71                 before moving to the Summary & Confirmation section"
72             Exit Sub
73         End If
74         StopCalc
75         LastEmpRow = .Range("AB999").End(xlUp).Row 'Last Row of Employees
76         .Range("BC9:BC310").ClearContents 'Clear Any Existing Employees
77         .Range("BC9:BC" & LastEmpRow).Value = .Range("AB9:AB" & LastEmpRow).Value 'Copy Over
78         Employee ID's
79         .Range("BD6").Value = Sheet18.Range("C" & .Range("B3").Value).Value 'Payroll Date
80         .Range("BL3").Value = Sheet18.Range("F" & .Range("B3").Value).Value 'Payroll Status
81
82         If .Range("BL3").Value = "Pending Confirmation" Then
83             .Shapes("PaySummConfirmedGrp").Visible = msoFalse
84             .Shapes("PaySummPendingGrp").Visible = msoCTrue
85         Else:
86             .Shapes("PaySummConfirmedGrp").Visible = msoCTrue
87             .Shapes("PaySummPendingGrp").Visible = msoFalse
88         End If
89         .Range("BG6").Value = .Range("J6").Value 'From Date
90         .Range("BI6").Value = .Range("J8").Value 'To Date
91         .Range("C:AY").EntireColumn.Hidden = True
92         .Range("AZ:BY").EntireColumn.Hidden = False
93         .Shapes("PayDetailGrp").Visible = msoFalse
94         .Shapes("PayDetailOptions").Visible = msoFalse
95         .Shapes("PayReviewGrp").Visible = msoFalse
96         ActiveWindow.ScrollColumn = 52
97
98         Payroll_LoadPayrollSummary
99         ResetCalc
100    End With
101 End Sub

```

## A

ActiveWindow, [129](#), [130](#)

## C

ClearContents, [129](#), [130](#)

## E

Empty, [129](#)  
EntireColumn, [129](#), [130](#)  
Explicit, [129](#)

## H

Hidden, [129](#), [130](#)

## L

LastEmpRow, [129](#), [130](#)

## M

MsgBox, [129](#), [130](#)  
msoCTrue, [129](#), [130](#)  
msoFalse, [129](#), [130](#)  
msoTrue, [129](#)

## P

Payroll\_Details\_To\_Review, [129](#)  
Payroll\_LoadEmpReview, [129](#)  
Payroll\_LoadPayrollSummary, [130](#)  
Payroll\_SelectDetailsSection, [129](#)  
Payroll\_SelectReviewSection, [129](#)  
Payroll\_SelectSummarySection, [130](#)  
PayrollRow, [129](#)

## R

Range, [129](#), [130](#)  
ResetCalc, [129](#), [130](#)  
Row, [129](#), [130](#)

## S

ScrollColumn, [129](#), [130](#)  
Shapes, [129](#), [130](#)  
Sheet16, [129](#), [130](#)  
Sheet18, [129](#), [130](#)  
StopCalc, [129](#), [130](#)

## V

Value, [129](#), [130](#)  
Visible, [129](#), [130](#)

## X

xLUp, [129](#), [130](#)

```
Option Explicit
```

```
Sub Payroll_AddSelectedEmpl()
```

```
Dim PaidRow As Long
```

```
Dim SelRow As Long
```

```
Dim LastPaidRow As Long
```

```
Dim LastSelRow As Long
```

```
Dim FoundEmpIDRng As Range
```

```
With Sheet16
```

```
StopCalc
```

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

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

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

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

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

```
'Check for Duplicates
```

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

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

```
LastPaidRow = LastPaidRow + 1
```

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

```
End If
```

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

```
End If
```

```
Next SelRow
```

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

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

```
With Sheet16.Sort
```

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

```
.Apply
```

```
End With
```

```
ResetCalc
```

```
End With
```

```
End Sub
```

```
Sub Payroll_EmpToPay_SelectAll()
```

```
Dim LastSelRow As Long
```

```
With Sheet16
```

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

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

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

```
End With
```

```
End Sub
```

```
Sub Payroll_FilterEmployees()
```

```
Dim LastRow As Long
```

```
Dim LastFiltRow As Long
```

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

```
With Sheet16
```

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

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

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

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

```
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("A9999" ).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("BF99999" ).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, [132](#), [133](#)  
AdvancedFilter, [133](#)  
Apply, [132](#), [133](#)

**C**

Clear, [132](#), [133](#)  
ClearContents, [132](#), [133](#)  
CopytoRange, [133](#)  
CriteriaRange, [133](#)

**D**

DataOption, [132](#), [133](#)

**E**

Explicit, [132](#)

**F**

Find, [132](#), [133](#)  
FoundEmpIDRng, [132](#), [133](#)

**K**

Key, [132](#), [133](#)

**L**

LastFiltRow, [132](#), [133](#)  
LastPaidRow, [132](#), [133](#)  
LastRow, [132](#), [133](#)  
LastSelRow, [132-134](#)

**N**

NoData, [133](#)

**O**

Order, [132](#), [133](#)

**P**

PaidRow, [132](#), [133](#)  
Payroll\_AddSelectedEmpl, [132](#)  
Payroll\_EmpToPay\_SelectAll, [132](#)  
Payroll\_FilterEmployees, [132](#)  
Payroll\_RemoveSelectedEmpl, [133](#)  
Payroll\_SelToPay\_SelectAll, [133](#)

**R**

Range, [132-134](#)  
ResetCalc, [132](#), [133](#)  
Row, [132-134](#)

**S**

SelRow, [132](#), [133](#)  
SetRange, [132](#), [133](#)  
Sheet16, [132-134](#)  
Sheet2, [132](#), [133](#)  
Sort, [132](#), [133](#)  
SortFields, [132](#), [133](#)  
SortOn, [132](#), [133](#)  
StopCalc, [132](#), [133](#)

**U**

Unique, [133](#)

**V**

Value, [132-134](#)

**X**

xlAscending, [132](#), [133](#)  
xlFilterCopy, [133](#)  
xlSortNormal, [132](#), [133](#)

xlSortOnValues, [132](#), [133](#)

xlUp, [132-134](#)

xlValues, [132](#), [133](#)

xlWhole, [132](#), [133](#)

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

## C

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

## E

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

## F

Formula, [136](#)

## L

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

## M

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

## N

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

## P

Payroll\_ConfirmCreate, [136](#)  
Payroll\_LoadPayrollSummary, [136](#)  
Payroll\_PrintPayStubs, [137](#)  
PayrollRow, [136](#)

## R

Range, [136](#), [137](#)  
Reports\_PrintOut, [137](#)  
Row, [136](#), [137](#)

## S

Shapes, [136](#)  
Sheet15, [137](#)  
Sheet16, [136](#), [137](#)  
Sheet18, [136](#)  
Sheet20, [136](#)

## U

Unique, [136](#)

## V

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

## X

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

```
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

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
```

—, 150

## A

Activate, 142  
 ActiveCell, 141, 143-149  
 ActiveSheet, 139-152  
 ActiveWorkbook, 141, 142  
 ActSht, 141, 142  
 Add, 141, 142  
 Address, 141, 142  
 After, 141  
 Array, 139-141, 149-152  
 AutoFill, 142, 143

## B

Bold, 140, 141, 151

## C

CalCol, 141, 142  
 CalCol1, 139  
 CalCol2, 139  
 CalCol3, 139  
 CalCol4, 139  
 CalCol5, 139  
 CalCol6, 139  
 CalCol7, 139  
 CalCol8, 139  
 CalCol9, 140  
 CalendarHide, 140  
 CalendarShow, 140  
 CalFormulaReplacement, 140, 141  
 Cells, 141, 142  
 CheckForSheet, 141  
 Chr, 142, 143  
 ColCnt, 141, 142  
 ColNum, 141  
 Copy, 141  
 Count, 141  
 CreateCalSht, 141  
 CreateWS, 141

## D

DayBtn1, 143  
 DayBtn10, 143  
 DayBtn11, 144  
 DayBtn12, 144  
 DayBtn13, 144  
 DayBtn14, 144  
 DayBtn15, 144  
 DayBtn16, 144  
 DayBtn17, 144  
 DayBtn18, 145  
 DayBtn19, 145  
 DayBtn2, 145  
 DayBtn20, 145  
 DayBtn21, 145  
 DayBtn22, 145  
 DayBtn23, 145  
 DayBtn24, 146  
 DayBtn25, 146  
 DayBtn26, 146  
 DayBtn27, 146  
 DayBtn28, 146  
 DayBtn29, 146  
 DayBtn3, 146  
 DayBtn30, 147  
 DayBtn31, 147  
 DayBtn32, 147

DayBtn33, 147  
 DayBtn34, 147  
 DayBtn35, 147  
 DayBtn36, 148  
 DayBtn37, 148  
 DayBtn38, 148  
 DayBtn39, 148  
 DayBtn4, 148  
 DayBtn40, 148  
 DayBtn41, 148  
 DayBtn42, 149  
 DayBtn5, 149  
 DayBtn6, 149  
 DayBtn7, 149  
 DayBtn8, 149  
 DayBtn9, 149  
 DayCnt, 141, 142  
 DayName, 139, 140, 151  
 DayNum, 140, 141  
 Destination, 142, 143  
 DrawingObject, 140-142

## E

Empty, 140, 143-149, 151

## F

Fill, 139, 140, 151  
 Font, 140, 141, 151  
 ForeColor, 139, 140, 151  
 Formula, 140-142

## G

Group, 149, 150  
 GroupCal, 140, 143, 149

## I

InStr, 140, 150  
 IsDate, 140

## L

Left, 141, 150  
 Len, 150

## M

MacroLink, 150  
 MacroLinkRemover, 140, 150  
 Month, 140  
 MsgBox, 141, 143  
 msoCTrue, 141, 150  
 msoFalse, 140, 143-149, 151  
 msoTrue, 141

## N

Name, 141, 142, 149, 150  
 Names, 142  
 NewLink, 150, 151  
 NextMonth, 151  
 NextShp, 150, 151  
 NoCal, 140, 141, 143  
 NumberFormat, 143

## O

Offset, 141, 143-149  
 OnAction, 140, 142, 150, 151

## P

Paste, 142  
 Placement, 150  
 PrevMonth, 151

## R

Range, 139-152  
 RefersTo, 142  
 RGB, 139, 140, 151  
 Right, 150  
 RowCnt, 141, 142  
 RowNum, 141

## S

SelCell, 139-141  
 Selection, 139, 140, 149, 150  
 Shape, 141, 150  
 ShapeRange, 139, 140, 149, 150  
 Shapes, 139-152  
 Sheets, 139-149, 151  
 ShowSettings, 151  
 shp, 141, 150, 151  
 Split, 150  
 SplitLink, 150

## T

TextFrame2, 140, 141, 151  
 TextRange, 140, 141, 151  
 ThisWorkbook, 141  
 Top, 141

## U

Ungroup, 152  
 UngroupCal, 140, 142, 152

## V

Value, 139-149, 151  
 Visible, 140-152

## W

Worksheet, 141  
 ws, 141

## X

xlFillDefault, 142, 143  
 xlMove, 150  
 xlSheetHidden, 142

## Y

Year, 140

```
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, [154](#)

**C**

Clear, [154](#)

ClearFormats, [154](#)

ClearRng, [154](#)

**E**

Explicit, [154](#)

**L**

LastRow, [154](#)

Locked, [154](#)

**P**

PageSetup, [154](#)

PrintArea, [154](#)

PrintOut, [154](#)

PrntRng, [154](#)

**R**

Range, [154](#)

Reports\_Clear, [154](#)

Reports\_PrintOut, [154](#)

Row, [154](#)

**S**

Sheet15, [154](#)

**U**

UnMerge, [154](#)

**X**

xUp, [154](#)

```

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
    format
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
    'Item Name
41 On Error Resume Next
42 .Range("AB" & ReportRow).Value = 1 / (TimeInteger / Sheet19.Range("S" &
    PayrollItemRow).Value) 'Hours
43 On Error GoTo 0
44 .Range("AD" & ReportRow).Value = Sheet19.Range("R" & PayrollItemRow).Value
    'Amount
45 'Copy Formulas
46 .Range("AA" & ReportRow & ":AE" & ReportRow).Copy
47 ReportRow = ReportRow + 1
48 .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormats
49 .Range("AA" & ReportRow & ":AE" & ReportRow).PasteSpecial xlPasteFormulas
50 .Range("AA" & ReportRow & ":AB" & ReportRow).ClearContents
51 .Range("AD" & ReportRow).ClearContents
52 End If
53 Next PayrollItemRow
54
55 'Add in Overtime (if any) - 2nd Loop
56 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, 156  
Application, 158

**C**

Clear, 156  
ClearContents, 156-158  
Copy, 156-158  
CopytoRange, 156  
CriteriaRange, 156  
CutCopyMode, 158

**E**

EmplID, 156  
Empty, 156  
Explicit, 156

**L**

LastPayrollItemFilterRow, 156  
LastPayrollItemRow, 156-158  
LastReportRow, 156, 158

**N**

NoPayData, 156, 158

**P**

PasteSpecial, 156-158  
PayID, 156  
PayrollItemRow, 156-158  
PayrollRow, 156  
PayType, 156-158

**R**

Range, 156-158  
ReportRow, 156-158  
Reports\_EarningsStatement, 156  
ResetCalc, 158  
Row, 156  
RptRng, 156, 158

**S**

Sheet15, 156  
Sheet19, 156-158  
Sheet4, 156  
StopCalc, 156

**T**

TimeInteger, 156, 157

**U**

Unique, 156

**V**

Value, 156-158

**X**

xlFilterCopy, 156  
xlPasteAll, 157, 158  
xlPasteColumnWidths, 158  
xlPasteFormats, 156-158  
xlPasteFormulas, 156-158  
xlUp, 156

```

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, [160](#)  
Application, [160](#)

## C

Calculate, [160](#)  
ClearContents, [160](#)  
Copy, [160](#)  
CopytoRange, [160](#)  
CriteriaRange, [160](#)  
CutCopyMode, [160](#)

## E

Empty, [160](#)  
Explicit, [160](#)

## L

LastFilterRow, [160](#)  
LastReportRow, [160](#)  
LastTimeclockRow, [160](#)

## N

NoTimeData, [160](#)

## P

PasteSpecial, [160](#)

## R

Range, [160](#)  
Report\_TimelockReport, [160](#)  
ResetCalc, [160](#)  
Row, [160](#)  
RptRng, [160](#)

## S

Sheet15, [160](#)  
Sheet9, [160](#)  
StopCalc, [160](#)

## U

Unique, [160](#)

## V

Value, [160](#)

## X

xlFilterCopy, [160](#)  
xlPasteAll, [160](#)  
xlPasteColumnWidths, [160](#)  
xlPasteFormulas, [160](#)  
xlUp, [160](#)

```
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, 162  
ActiveCell, 162  
Add, 163  
Application, 163, 164

**B**

Before, 163

**C**

Caption, 163, 164  
Cells, 162  
Column, 162  
Columns, 162  
CommandBarButton, 163  
CommandBarPopup, 163  
CommandBars, 163, 164  
Controls, 163, 164  
Count, 162  
CreateLeaveEvent, 162, 163  
CreateLeaveEvent10, 162  
CreateLeaveEvent11, 162  
CreateLeaveEvent12, 162  
CreateLeaveEvent13, 162  
CreateLeaveEvent14, 162  
CreateLeaveEvent15, 162  
CreateLeaveEvent16, 162  
CreateLeaveEvent17, 162  
CreateLeaveEvent18, 162  
CreateLeaveEvent19, 163  
CreateLeaveEvent20, 163  
CreateLeaveEvent21, 163  
CreateLeaveEvent22, 163  
CreateLeaveEvent23, 163  
CreateLeaveEvent7, 163  
CreateLeaveEvent8, 163  
CreateLeaveEvent9, 163

**D**

DateRow, 162  
Delete, 163, 164

**E**

EditTimeClockBtn, 163, 164  
EmpRow, 162  
Event\_AddNew, 162  
Explicit, 162

**F**

FacId, 163, 164

**G**

GoToEventBtn, 163, 164

**L**

LeaveBtn, 163  
LeaveTypeRow, 163, 164

**M**

msoControlButton, 163  
msoControlPopup, 163

**O**

OnAction, 164

**R**

Range, 162, 163  
RightClick\_CallEvents, 163  
RightClick\_Cancel, 164

Row, 162

**S**

SelCol, 162  
SelColumns, 162  
Selection, 162  
SelLeaveRow, 162, 163  
SelRow, 162  
Sheet2, 162  
Sheet4, 162, 163  
Sheet5, 162  
Sheet8, 162

**V**

Value, 162, 163



```

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

```

1 2 3

```

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 Event_Load 'Run Load Macro Event
120 End With
121 End Sub
122
123 Sub Sched_MonthView()
124 With Sheet8
125 StopCalc
126 .Shapes("MonthGrp" ).Visible = msoCTrue
127 .Shapes("WeekGrp" ).Visible = msoFalse
128 .Range("F:CZ" ).EntireColumn.Hidden = False
129 .Range("DA:F0" ).EntireColumn.Hidden = True
130 ActiveWindow.FreezePanes = False
131 Range("F8" ).Select
132 ActiveWindow.FreezePanes = True
133
134 ResetCalc
135 End With
136 End Sub
137
138 Sub Sched_NextMonth()
139 Sched_ClearFilters
140 With Sheet8
141 .Range("F8:AK314" ).ClearComments 'Clear All Comments
142 If .Range("B1" ).Value = 12 Then
143 .Range("E2" ).Value = .Range("E2" ).Value + 1
144 .Range("B1" ).Value = 1
145 Else: .Range("B1" ).Value = .Range("B1" ).Value + 1
146 End If
147 End With
148 Sched_Refresh
149 End Sub
150
151 Sub Sched_NextWeek()
152 Sched_ClearFilters
153 With Sheet8
154 .Range("B2" ).Value = .Range("B2" ).Value + 7
155 If Year(.Range("B2" ).Value) <> .Range("E2" ).Value Then .Range("E2" ).Value = .Range(
156 "E2" ).Value + 1
157 .Range("F8:DG314" ).ClearComments 'Clear All Comments
158 Sched_Refresh
159 End With
160 End Sub
161
162 Sub Sched_PrevMonth()
163 Sched_ClearFilters
164 With Sheet8
165 .Range("F8:AK314" ).ClearComments 'Clear All Comments
166 If .Range("B1" ).Value = 1 Then
167 .Range("E2" ).Value = .Range("E2" ).Value - 1
168 .Range("B1" ).Value = 12
169 Else: .Range("B1" ).Value = .Range("B1" ).Value - 1
170 End If
171 End With
172 Sched_Refresh
173 End Sub
174
175 Sub Sched_PrevWeek()
176 With Sheet8
177 .Range("B2" ).Value = .Range("B2" ).Value - 7

```

1 2

```
1 2
177 If Year(.Range("B2" ).Value) <> .Range("E2" ).Value Then .Range("E2" ).Value = .Range(
178 "E2" ).Value - 1
179 .Range("F8:DG314" ).ClearComments 'Clear All Comments
180 Sched_Refresh
181 End With
182 End Sub
183 Sub Sched_Refresh()
184 With Sheet8
185 StopCalc
186 .Range("F6:F06" ).Copy
187 .Range("F8:F0314" ).PasteSpecial xlPasteFormulas
188 .Calculate
189 .Range("F8:F0314" ).Copy
190 .Range("F8:F0314" ).PasteSpecial xlPasteValues
191 Application.CutCopyMode = False
192 .Range("E2" ).Select
193 .Range("F8:DG314" ).ClearComments 'Clear All Comments
194 ResetCalc
195 End With
196 End Sub
197 Sub Sched_ScrollToRow()
198 ActiveWindow.ScrollRow = Sheet8.Range("B3" ).Value
199 End Sub
200 Sub Sched_ShowHideCapacity()
201 If Sheet8.Range("B7" ).Value = True Then Sheet8.Range("7:7" ).EntireRow.Hidden = False
202 Else: Sheet8.Range("7:7" ).EntireRow.Hidden = True
203 End Sub
204 Sub Sched_ThisMonth()
205 Sched_ClearFilters
206 Sheet8.Range("E2" ).Value = Year(Now)
207 Sheet8.Range("B1" ).Value = Month(Now)
208 Sched_Refresh
209 End Sub
210 Sub Sched_ThisWeek()
211 Sched_ClearFilters
212 Sheet8.Range("B2" ).Value = Sheet8.Range("A11" ).Value
213 Sheet8.Range("E2" ).Value = Year(Date)
214 Sched_Refresh
215 End Sub
216 Sub Sched_WeekView()
217 With Sheet8
218 StopCalc
219 .Shapes("MonthGrp" ).Visible = msoFalse
220 .Shapes("WeekGrp" ).Visible = msoCTrue
221 .Range("F:CZ" ).EntireColumn.Hidden = True
222 .Range("DA:F0" ).EntireColumn.Hidden = False
223 ActiveWindow.FreezePanes = False
224 Range("DA8" ).Select
225 ActiveWindow.FreezePanes = True
226 .Range("F8:DG309" ).ClearComments 'Clear All Comments
227 ResetCalc
228 End With
229 End Sub
```

→, 166

## A

ActCol, 166, 167  
 Activate, 168  
 ActiveCell, 166, 167  
 ActiveSheet, 167  
 ActiveWindow, 168, 169  
 ActRow, 166, 167  
 AddComment, 166, 167  
 Application, 169  
 AutoFilter, 167  
 AutoFilterMode, 167  
 AutoShapeType, 166, 167

## B

BackColor, 166, 167  
 Bold, 166, 167

## C

Calculate, 169  
 Cells, 166, 167  
 Characters, 166, 167  
 ClearComments, 166-169  
 ColorIndex, 166, 167  
 Column, 166, 167  
 Comment, 166, 167  
 Copy, 169  
 CutCopyMode, 169

## E

Empty, 166, 167  
 EndOn, 166  
 EntireColumn, 168, 169  
 EntireRow, 169  
 Event\_Load, 168  
 EventComment, 166  
 EventCreatedOn, 166  
 EventEmpl, 166  
 EventID, 166-168  
 EventIDRng, 166  
 EventName, 166  
 EventNotes, 166  
 EventRecFreq, 166  
 EventRecQty, 166  
 EventRecur, 166  
 EventRow, 166  
 EventType, 166  
 Explicit, 166

## F

Fill, 166, 167  
 Find, 166, 167  
 Font, 166, 167  
 ForeColor, 166, 167  
 FoundEventIDRng, 166  
 FoundHolDtRng, 167  
 FreezePanels, 168, 169

## H

Height, 166, 167  
 Hidden, 168, 169  
 HolDtRng, 167  
 HolidayDt, 167  
 HolidayName, 167  
 HolidayRow, 167

## L

Line, 166, 167

## M

Month, 169  
 MsgBox, 167  
 msoCTrue, 168, 169  
 msoFalse, 168, 169  
 msoGradientDiagonalUp, 166, 167  
 msoShapeRoundedRectangle, 166, 167  
 msoTrue, 166, 167

## N

Name, 166, 167  
 Now, 169

## O

OneColorGradient, 166, 167

## P

PasteSpecial, 169

## R

Range, 166-169  
 ResetCalc, 168, 169  
 RGB, 166, 167  
 Row, 166, 167

## S

Sched\_AddComment\_Event, 166  
 Sched\_AddComment\_Holiday, 167  
 Sched\_ClearFilters, 167-169  
 Sched\_GoToEvent, 167  
 Sched\_MonthView, 168  
 Sched\_NextMonth, 168  
 Sched\_NextWeek, 168  
 Sched\_PrevMonth, 168  
 Sched\_PrevWeek, 168  
 Sched\_Refresh, 168, 169  
 Sched\_ScrollToRow, 169  
 Sched\_ShowHideCapacity, 169  
 Sched\_ThisMonth, 169  
 Sched\_ThisWeek, 169  
 Sched\_WeekView, 169  
 ScrollRow, 169  
 Shape, 166, 167  
 Shapes, 168, 169  
 Sheet4, 167  
 Sheet5, 168  
 Sheet7, 166  
 Sheet8, 166-169  
 ShowAllData, 167  
 Size, 166, 167  
 StartOn, 166  
 StopCalc, 168, 169

## T

text, 166, 167  
 TextFrame, 166, 167

## V

Value, 166-169  
 vbCrLf, 166  
 Visible, 166-169

## W

Width, 166, 167

## X

xlPasteFormulas, 169  
 xlPasteValues, 169  
 xlValues, 166, 167  
 xlWhole, 166, 167

## Y

Year, 168, 169

```
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



```

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     "05Msel" , "10Msel" , "15Msel" , "20Msel" , "25Msel" , "30Msel" , "35Msel" ,
    "40Msel" , _
238     "45Msel" , "50Msel" , "55Msel" , "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
```



—

\_, 174, 175

**A**

ActiveCell, 171, 172, 180

ActiveSheet, 171-181

AMPMSel, 171

AMSel, 171

Application, 173, 174

Array, 171, 172, 174, 179, 180

**C**

Characters, 171-178, 180

CurLeft, 172-174

CurTop, 172-174

**D**

DoneBtn, 171

**E**

Empty, 172, 180

ExitSub, 171, 172, 174, 175

Explicit, 171

**F**

Fill, 172, 174, 178-180

Font, 172, 174

ForeColor, 172, 174, 178-180

Format, 180

**G**

Group, 171, 172, 175

GrpHrs, 171, 180

GrpMin, 171, 180

GrpRange, 171, 172, 174, 175

GrpSet, 171, 180

GrpShapes, 171, 172, 174, 175

**H**

Height, 173, 174

HideHrSel, 172, 174-177, 180

HideMinSel, 172, 177, 178, 180

HoursDisplay, 172, 180

Hr, 180

**I**

InStr, 173, 180

Inter, 173, 174

Intg10, 173, 174

Intg15, 173, 174

Intg5, 173, 174

**L**

Left, 173, 174, 180

Len, 173

**M**

MacroLink, 173

MacroLinkRemover, 173, 180

MinDisplay, 173, 175-177, 179

MsgBox, 180

msoCTrue, 178-180

msoFalse, 179, 180

**N**

Name, 171, 172, 175

NewLink, 173

NextShp, 173

NoTP, 180

Now, 173, 174

**O**

Offset, 180

OnAction, 173, 180

**P**

Placement, 175

PMSel, 171, 174

ProtTP, 171, 173-180

**R**

Range, 171, 172, 174, 179, 180

RGB, 172, 174, 178-180

Right, 173, 180

**S**

SelCell, 180

Select1, 175

Select10, 175

Select11, 175

Select12, 175

Select2, 175

Select3, 176

Select4, 176

Select5, 176

Select6, 176

Select7, 176

Select8, 176

Select9, 176

Selection, 180

SelectM00, 177, 179

SelectM05, 177

SelectM10, 177

SelectM15, 177

SelectM20, 177

SelectM25, 177

SelectM30, 178

SelectM35, 178

SelectM40, 178

SelectM45, 178

SelectM50, 178

SelectM55, 178

Set10MinInt, 178

Set15MinInt, 179

Set5MinInt, 179

SetHideShow, 179

SetThm1, 179

SetThm2, 179

SetThm3, 179

SetThm4, 179

SetThm5, 180

Shape, 173

Shapes, 171-181

shp, 173

Siz, 172-174

Split, 173

SplitLink, 173

**T**

text, 171-178, 180

TextFrame, 171-178, 180

TextFrame2, 172, 174

TextRange, 172, 174

Top, 173, 174, 180

TPShow, 180

**U**

Ungroup, 180, 181

UnGrpHrs, 180

UnGrpMin, 180, 181

UnGrpSet, 180, 181

UnProtTP, 171-181

**V**

Value, 171, 172, 180

Visible, 171-180

**W**

Wait, 173, 174

Width, 173, 174

```

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, 184  
ActiveSheet, 183, 184

**B**

BreakIn, 183-185  
BreakInVal, 183, 184  
BreakOut, 183-185  
BreakOutVal, 183, 184

**C**

Caption, 183, 185  
Cells, 184  
ClockIn, 183-185  
ClockInVal, 183, 184  
ClockOut, 183-185  
ClockOutVal, 183-185  
CodeName, 183, 184  
Column, 184

**D**

DateRow, 184

**E**

EditTimeClock, 183-185  
EditTimeClock\_Calculate, 183, 185  
EditTimeClock\_Save, 183  
Empty, 183, 184  
Explicit, 183

**F**

Format, 183, 184

**H**

Hide, 184

**I**

Int, 184

**M**

MsgBox, 183

**N**

Notes, 184, 185  
Now, 184

**O**

OTHrsValue, 183, 185

**P**

Payroll\_LoadEmpReview, 184

**R**

Range, 183-185  
RegHrsValue, 183, 185  
Row, 183, 184

**S**

Sched\_Refresh, 184  
SelCol, 184  
SelRow, 183, 184  
Sheet14, 184  
Sheet16, 183  
Sheet4, 183  
Sheet8, 183, 184  
Sheet9, 183-185  
Show, 185

**T**

TCRow, 183-185  
TimeClock\_EditEntry, 184

**V**

Value, 183-185  
vbNullString, 185

**X**

xLUp, 183

```

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 Exit Sub
210 EntryAddedAlready:
211 Timeclock_AlreadyEntered
212 End Sub
213
214 Sub Timeclock_IDNotFound()
215 Dim i As Integer
216 Dim delay As Double
217 Dim StartTime As Double
218 With ActiveSheet.Shapes("IDNotFound" )

```

1 2

```
1 2
219 .Visible = True
220 For i = 1 To 300
221     .Fill.Transparency = i / 300
222     delay = 0.00004
223     StartTime = Timer
224     Do
225         DoEvents
226     Loop While Timer - StartTime < delay
227 Next i
228 .Visible = False
229 End With
230 End Sub
231 Sub Timeclock_StartStop()
232     With Sheet12
233         .Range("J13").Select
234         RunClk = Not (RunClk)
235         Do While RunClk = True
236             DoEvents
237             .Range("A29") = TimeValue(Now)
238             DoEvents
239             .Range("A30") = Now()
240         Loop
241     End With
242 End Sub
243
244
```

**A**

ActiveSheet, 187, 190  
Application, 188, 190

**B**

BreakThreshold, 188, 189

**C**

ClearContents, 188, 190

**D**

Day, 188  
DayThreshold, 188, 189  
delay, 187, 190, 191  
Delete, 187  
DoEvents, 187, 191

**E**

EmpID, 187-189  
EmpName, 187-190  
EmpRow, 187, 188, 190  
Empty, 188-190  
EntryAdded, 188-190  
EntryAddedAlready, 190  
EntryType, 187-190  
Explicit, 187

**F**

Fill, 187, 191  
Find, 188  
FoundPrevTimeClockIDRng, 188  
FoundTimeClockIDRng, 187-189

**H**

Height, 187

**I**

i, 187, 190, 191  
IncrementLeft, 187  
IncrementTop, 187  
Insert, 187

**L**

Left, 187  
LockAspectRatio, 187

**M**

MessRow, 187, 190  
Month, 188  
msoCTrue, 190  
msoFalse, 188, 190  
msoSoftEdgeType4, 187  
msoTrue, 187

**N**

Name, 187  
NoPic, 187  
Now, 188-191

**P**

PicPath, 187  
Pictures, 187

**R**

Range, 187-191  
Replace, 190  
Round, 190  
Row, 188, 189  
RunClk, 187, 190, 191

**S**

ScreenUpdating, 190  
ShapeRange, 187  
Shapes, 187, 188, 190  
Sheet12, 187, 188, 190, 191  
Sheet2, 187, 190  
Sheet4, 188-190  
Sheet9, 188, 190  
SoftEdge, 187  
StartTime, 187, 190, 191

**T**

text, 188  
Timeclock\_AlreadyEntered, 187, 190  
Timeclock\_DisplayPicture, 187, 190  
Timeclock\_EnterID, 187  
Timeclock\_IDNotFound, 188, 190  
Timeclock\_StartStop, 191  
TimeClockID, 187-189  
TimeClockID\_Prev, 187, 188  
TimeClockIDRng, 187, 188  
TimeClockRow, 187-190  
Timer, 187, 191  
TimeRow, 187, 190  
TimeValue, 190, 191  
Top, 187  
TotalBreakHours, 188, 190  
TotalHours, 188, 190  
Transparency, 187, 191

**V**

Value, 187-190  
Visible, 187, 188, 190, 191

**W**

Wait, 190  
WorksheetFunction, 188

**X**

xlUp, 189  
xlValues, 188  
xlWhole, 188

**Y**

Year, 188

```

1  Option Explicit
2
3  Sub TimeClock_SyncRemoteEntries()
4      Dim FSO As Object
5      Dim TimeClockRow As Long
6      Dim TimeClockFolder As String
7      Dim FileName As String
8      Dim TCFile As String
9      Dim TCText As String
10     Dim TCTextLine As String
11     Dim TimeClockID As String
12     Dim EmpID As String
13     Dim ClockIn As String
14     Dim BreakOut As String
15     Dim BreakIn As String
16     Dim ClockOut As String
17     Dim TimeClockIDRng As Range
18     Dim FoundTimeClockIDRng As Range
19     Dim EmpIDRng As Range
20     Dim FoundEmpIDRng As Range
21     Set TimeClockIDRng = Sheet9.Range("TimeClockList_ID" )
22     Set EmpIDRng = Sheet2.Range("EmployeeID" )
23     With Sheet9
24         If Sheet4.Range("RemoteTimeClockFolder" ).Value = Empty Then
25             Sheet4.Activate
26             Sheet4.Range("H4" ).Select
27             MsgBox "Please make sure to select a default Dropbox (or shared) folder in which
28                 remote Time Clock Entries are sent"
29             BrowseRemoteTimeClockFolder
30             If Sheet4.Range("RemoteTimeClockFolder" ).Value = Empty Then Exit Sub 'If there is
31                 still no folder then exit sub, otherwise continue
32             End If
33             TimeClockFolder = Sheet4.Range("RemoteTimeClockFolder" ).Value 'Set Time Clock Folder
34             Set FSO = CreateObject("Scripting.FileSystemObject" ) 'Set File System Object
35             If Dir(TimeClockFolder & "/Time Clock Archives/" , vbDirectory) = "" Then FSO.
36                 createfolder (TimeClockFolder & "/Time Clock Archives/" )
37
38             FileName = Dir(TimeClockFolder & "\*.txt" )
39
40             Do While Len(FileName) > 0 'Start of Loop
41                 TCFile = TimeClockFolder & "\" & FileName
42
43                 Open TCFile For Input As #1
44                 Line Input #1, TCText
45                 Close #1
46
47                 EmpID = Mid(TCText, 13, InStr(TCText, "Clock In:" ) - 13) 'Starting Point of
48                 Employee ID
49                 ClockIn = Mid(TCText, InStr(TCText, "Clock In:" ) + 9, InStr(TCText, "Break Out:" )
50                     - (InStr(TCText, "Clock In:" ) + 9))
51                 BreakOut = Mid(TCText, InStr(TCText, "Break Out:" ) + 10, InStr(TCText, "Break In:" )
52                     - (InStr(TCText, "Break Out:" ) + 10))
53                 BreakIn = Mid(TCText, InStr(TCText, "Break In:" ) + 9, InStr(TCText, "Clock Out:" )
54                     - (InStr(TCText, "Break In:" ) + 9))
55                 ClockOut = Mid(TCText, InStr(TCText, "Clock Out:" ) + 10, (Len(TCText) - (InStr(
56                     TCText, "Clock Out:" ) + 10)))
57                 TimeClockID = Application.WorksheetFunction.text(ClockIn, "yyyy-mm-dd-" ) & EmpID
58
59                 Set FoundTimeClockIDRng = TimeClockIDRng.Find(TimeClockID, , xlValues, xlWhole)
60                 If Not FoundTimeClockIDRng Is Nothing Then TimeClockRow = FoundTimeClockIDRng.Row
61                 Else: TimeClockRow = .Range("A99999" ).End(xlUp).Row + 1 'Use Existing Found Row
62             End While
63         End With
64     End Sub

```

1 2 3

```
1 2 3
or First Available Row
53 .Range("A" & TimeClockRow).Value = TimeClockID 'Time Clock ID
54 .Range("B" & TimeClockRow).Value = EmpID 'Employee ID
55 Set FoundEmpIDRng = EmpIDRng.Find(EmpID, , xlValues, xlWhole)
56 If Not FoundEmpIDRng Is Nothing Then .Range("C" & TimeClockRow).Value = Sheet2.
Range("BC" & FoundEmpIDRng.Row).Value 'Add Employee If ID is Found
57 If .Range("D" & TimeClockRow).Value = Empty Then .Range("D" & TimeClockRow).Value
= ClockIn 'Add Clock In Only If Existing Value is Empty
58 If .Range("E" & TimeClockRow).Value = Empty Then .Range("E" & TimeClockRow).Value
= BreakOut 'Add Break Out Only If Existing Value is Empty
59 If .Range("F" & TimeClockRow).Value = Empty Then .Range("F" & TimeClockRow).Value
= BreakIn 'Add Break Out Only If Existing Value is Empty
60 If .Range("G" & TimeClockRow).Value = Empty Then .Range("G" & TimeClockRow).Value
= ClockOut 'Add Break Out Only If Existing Value is Empty
61 .Range("I" & TimeClockRow).Value = "=Row()" 'Set Row #
62 .Range("J" & TimeClockRow).Value = "Remote" 'Entry Type
63
64 'Move To Time Clock Archives Folder
65
66 FSO.MoveFile Source:=TCfile, Destination:=TimeClockFolder & "/Time Clock Archives/"
& FileName 'Move To Archives Folder
67 FileName = Dir() 'Clear Curent File Name
68 Loop 'End Of Loop
69 End With
70 End Sub
71
72
73
```

**A**

Activate, 193  
Application, 193

**B**

BreakIn, 193, 194  
BreakOut, 193, 194  
BrowseRemoteTimeClockFolder, 193

**C**

ClockIn, 193, 194  
ClockOut, 193, 194  
Close, 193  
createfolder, 193  
CreateObject, 193

**D**

Destination, 194  
Dir, 193, 194

**E**

EmpID, 193, 194  
EmpIDRng, 193, 194  
Empty, 193, 194  
Explicit, 193

**F**

FileName, 193, 194  
Find, 193, 194  
FoundEmpIDRng, 193, 194  
FoundTimeClockIDRng, 193  
FSO, 193, 194

**I**

Input, 193  
InStr, 193

**L**

Len, 193  
Line, 193

**M**

Mid, 193  
MoveFile, 194  
MsgBox, 193

**O**

Open, 193

**R**

Range, 193, 194  
Row, 193, 194

**S**

Sheet2, 193, 194  
Sheet4, 193  
Sheet9, 193  
Source, 194

**T**

TCTFile, 193, 194  
TCText, 193  
TCTextLine, 193  
text, 193  
TimeClock\_SyncRemoteEntries, 193  
TimeClockFolder, 193, 194  
TimeClockID, 193, 194  
TimeClockIDRng, 193  
TimeClockRow, 193, 194

**V**

Value, 193, 194  
vbDirectory, 193

**W**

WorksheetFunction, 193

**X**

xLUp, 193  
xlValues, 193, 194  
xlWhole, 193, 194

```
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
```

1 2

```
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 &HC00000
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.Show
108 End Sub
```



—  
\_, 196

## A

Activate, 197

## C

Calculate, 196  
Caption, 197  
CheckUser, 196  
ClearContents, 197  
CloseWorkbook, 197

## D

DrawMenuBar, 196, 197  
dwNewLong, 196

## E

Empty, 196  
Explicit, 196

## F

FindWindow, 196, 197  
frm, 197

## G

GetWindowLong, 196, 197

## H

Hide, 197  
HideBar, 197  
hWnd, 196  
hWndForm, 197

## L

LoginForm, 197  
LogOut, 197  
lpClassName, 196  
lpWindowName, 196

## M

Menu, 197  
MsgBox, 196

## N

Name, 197  
nIndex, 196

## P

PtrSafe, 196

## R

Range, 196, 197

## S

Save, 197  
SetWindowLong, 196, 197  
Sheet14, 196, 197  
Sheet14Activate, 197  
Show, 197  
ShowLoginForm, 197  
StartWorkbook, 197  
Style, 197

## T

ThisWorkbook, 197

## U

User\_SetAs\_Admin, 197

User\_SetAs\_Manager, 197  
User\_SetAs\_TimeClock, 197  
User\_SetAs\_User, 197  
UserType, 196, 197

## V

Value, 196, 197  
VBA7, 196  
Visible, 197

## W

WkSht, 197  
Worksheet, 197  
Worksheets, 197

## X

xlSheetVeryHidden, 197  
xlSheetVisible, 197

---

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, [199](#)

## E

Explicit, [199](#)

## N

Name, [199](#)

## R

Range, [199](#)

## S

Sheet1, [199](#)  
Sheet11, [199](#), [200](#)  
Sheet12, [199](#), [200](#)  
Sheet13, [199](#), [200](#)  
Sheet14, [199](#), [200](#)  
Sheet15, [199](#), [200](#)  
Sheet16, [199](#), [200](#)  
Sheet2, [199](#)  
Sheet3, [199](#)  
Sheet4, [199](#)  
Sheet5, [199](#)  
Sheet6, [199](#), [200](#)  
Sheet7, [199](#), [200](#)  
Sheet8, [199](#), [200](#)  
Sheet9, [199](#), [200](#)

## T

ThisWorkbook, [199](#)

## U

User\_SetAs\_Admin, [199](#)  
User\_SetAs\_Manager, [199](#)  
User\_SetAs\_TimeClock, [199](#)  
User\_SetAs\_User, [199](#)

## V

Visible, [199](#), [200](#)

## W

WkSht, [199](#)  
Worksheet, [199](#)  
Worksheets, [199](#)

## X

xlSheetHidden, [199](#), [200](#)  
xlSheetVeryHidden, [199](#), [200](#)  
xlSheetVisible, [199](#), [200](#)

# 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*