

VBA SOURCE CODE BOOK

EMPLOYEE MANAGER PT.15 CREATE PAYROLL PART 1



CREATE PAYROLL

1. Payroll Details 2. Review Employees 3. Summary & Create

3 of 11 Employees Reviewed (27.27%)

Prev. Employee Save & Next **Barret, Fredda 10/29/2018 - 11/30/2018**

Employees To Be Paid		Time Clock Summary		Pay Type: Hourly	
1002	Angle, Tanya	Total Reg. Hours: 36.43	Hourly Rate: \$15.00	Total Reg.:	\$546.50
1012	Batie, Theresa	Total OT Hours: 7.22	OT Rate: \$30.00	Total OT:	\$216.50

Employee Earnings Summary	
Total Earned Income:	\$763.00
Total Tax Deductions:	\$0.13
Other Deductions:	\$ 150.00
Total Additions:	\$ 175.00
Net Earnings:	\$787.87

Time Clock Period Detail					
Clock In	Clock Out-Break	Clock In-Break	Clock Out	Reg. Hrs	OT Hrs
11/21/18 9:33 AM			11/21/18 10:31 PM	8.00	4.97
11/16/18 7:12 PM			11/17/18 2:12 AM	7.00	
11/15/18 7:12 PM			11/16/18 1:12 AM	6.00	
11/14/18 5:50 PM	11/14/18 6:06 PM	11/14/18 6:15 PM	11/15/18 1:25 AM	7.43	
11/13/18 5:44 PM	11/14/18 1:47 AM	11/14/18 2:48 AM	11/14/18 5:00 AM	8.00	2.25

Federal & Local Tax Deductions		Other	
Tax Item	Amount	YTD	Deduction
Fed. Inc. Tax	\$0.01		Health Insurance
Fed Unemp. Ins.	\$0.09		Tools Payment
Social Security	\$0.03		

 *Excel For Freelancers*

How To Create A Full Payroll in Excel Pt.1 [Employee Manager Part 15]



[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



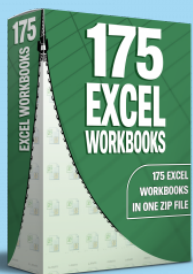
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



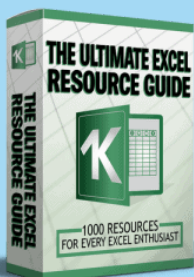
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[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
Sheet16.....	25
(Declarations).....	25
Worksheet_Change [Sub]	25
Worksheet_SelectionChange [Sub]	25
Sheet17.....	27
(Declarations).....	27
Sheet18.....	29
(Declarations).....	29
Sheet19.....	31
(Declarations).....	31
Sheet2.....	33
(Declarations).....	33
Sheet3.....	35
(Declarations).....	35
Worksheet_Change [Sub]	35
Sheet4.....	37
(Declarations).....	37
Worksheet_Activate [Sub].....	37
Worksheet_Change [Sub]	37
Worksheet_Deactivate [Sub].....	37
Worksheet_SelectionChange [Sub]	37
Sheet5.....	40
(Declarations).....	40
Worksheet_BeforeDoubleClick [Sub]	40
Worksheet_Change [Sub]	40
Worksheet_Deactivate [Sub].....	41
Worksheet_SelectionChange [Sub]	41
Sheet6.....	44
(Declarations).....	44
Worksheet_SelectionChange [Sub]	44
Sheet7.....	46
(Declarations).....	46
Sheet8.....	48
(Declarations).....	48
Worksheet_Activate [Sub].....	48
Worksheet_Change [Sub]	48
Worksheet_SelectionChange [Sub]	48
Sheet9.....	50
(Declarations).....	50
ThisWorkbook	52
(Declarations).....	52
Workbook_Activate [Sub].....	52
Workbook_BeforeClose [Sub]	52
Workbook_Deactivate [Sub].....	52

Workbook_SheetBeforeRightClick [Sub].....	52
Forms.....	54
CreatePayPeriod.....	54
(Declarations).....	54
CancelBtn_Click [Sub].....	54
ClsBtn_Click [Sub].....	54
CreateBtn_Click [Sub].....	54
EditTimeClock.....	56
(Declarations).....	56
BreakIn_AfterUpdate [Sub].....	56
BreakInNow_MouseDown [Sub].....	56
BreakOut_AfterUpdate [Sub].....	56
BreakOutNow_MouseDown [Sub].....	56
CancelBtn_MouseDown [Sub].....	56
ClockIn_AfterUpdate [Sub].....	56
ClockInNow_MouseDown [Sub].....	56
ClockOut_AfterUpdate [Sub].....	56
ClockOutNow_MouseDown [Sub].....	56
OkBtn_Click [Sub].....	56
UserForm_Activate [Sub].....	56
LoginForm.....	59
(Declarations).....	59
CloseBtn_Click [Sub].....	59
OkBtn_Click [Sub].....	59
UserForm_Initialize [Sub].....	59
Modules.....	61
Admin_EventNotif.....	61
(Declarations).....	61
Notif_AddEmailAttach [Sub].....	61
Notif_AddNew [Sub].....	61
Notif_CancelNew [Sub].....	61
Notif_Delete [Sub].....	61
Notif_Load [Sub].....	61
Notif_Save [Sub].....	61
Admin_Macros.....	64
(Declarations).....	64
BrowseEmployeeFolder [Sub].....	64
BrowseRemoteTimeClockFolder [Sub].....	64
CreatePayPeriods [Sub].....	64
ShowPayPeriodForm [Sub].....	65
Admin_Tabs.....	67
(Declarations).....	67
Admin_SwitchHorizontalTabs [Sub].....	67
AutoTab_Macros.....	69
do_nothing [Sub].....	69
GetColLtr [Function].....	69
GetTabOrder [Function].....	69
SetOnkey [Sub].....	69
TabRange [Sub].....	70
UpOrDownArrow [Sub].....	70
CodeReset_Macs.....	73
(Declarations).....	73
ResetCalc [Sub].....	73
StopCalc [Sub].....	73
Empl_Attach.....	75
(Declarations).....	75
Attach_Add [Sub].....	75
Attach_Delete [Sub].....	75
Attach_DisplayThumbnail [Sub].....	76
Attach_Filter [Sub].....	77
Attach_FilterClear [Sub].....	77
Attach_Open [Sub].....	77
Attach_Refresh [Sub].....	77
Attach_ShowAll [Sub].....	78
Empl_IDCard.....	80
(Declarations).....	80

EmplIDCard_CreateTextBox [Sub]	80
EmplIDCard_Customize [Sub]	80
EmplIDCard_FieldChange [Sub]	80
EmplIDCard_Hide [Sub]	80
EmplIDCard_HideCustom [Sub]	80
EmplIDCard_HidePic [Sub]	81
EmplIDCard_Print [Sub]	81
EmplIDCard_Show [Sub]	81
EmplIDCard_ShowPic [Sub]	81
Empl_LoadMacros	83
(Declarations).....	83
Empl_EventsRefresh [Sub]	83
Empl_LeaveRefresh [Sub]	83
Empl_Load [Sub]	83
Empl_PayItemsRefresh [Sub]	84
Empl_TimeclockRefresh [Sub]	85
Empl_WeeklySchedRefresh [Sub]	85
Empl_Misc.....	88
(Declarations).....	88
Empl_Archive [Sub]	88
Empl_CancelNew [Sub]	88
Empl_New [Sub]	88
Empl_ReActivate [Sub]	89
Empl_SaveNew [Sub]	89
Empl_SortByName [Sub]	90
Empl_PicsMacs.....	92
(Declarations).....	92
Add_EmplPic [Sub]	92
Show_EmplPic [Sub]	92
Empl_Tabs	95
(Declarations).....	95
Empl_SwitchHorizontalTabs [Sub]	95
EmpTab_Events [Sub]	95
EmpTab_GenInfo [Sub]	95
EmpTab_Leave [Sub]	95
EmpTab_PayDetails [Sub]	95
EmpTab_PayHistory [Sub]	95
EmpTab_Scheduling [Sub]	95
EmpTab_TimeClock [Sub]	96
SwitchVerticalTabs [Sub]	96
Events_Misc.....	98
(Declarations).....	98
DeleteFilters [Sub]	98
Event_AddEmailAttach [Sub]	98
Event_AddNew [Sub]	98
Event_AddNotifDefaults [Sub]	98
Event_CancelNew [Sub]	98
Event_Delete [Sub]	98
Event_Load [Sub]	99
Event_Refresh [Sub]	99
Event_SaveNew [Sub]	100
FadeOutMessage.....	102
(Declarations).....	102
EmployIDNotFound [Sub]	102
FolderPickMac.....	104
(Declarations).....	104
GetFolder [Sub]	104
MailToSelected [Sub]	104
Notification_Macros.....	106
(Declarations).....	106
Notification_Close [Sub]	106
Notifications_Check [Sub]	106
Notifications_Create [Sub]	106
testting [Sub]	108
Payroll_ReviewEmpMacs	110
(Declarations).....	110

Payroll_LoadEmpReview [Sub]	110
Payroll_SaveEmplPayMac	114
(Declarations).....	114
Payroll_SaveEmployeePay [Sub]	114
Payroll_SelectSections	118
(Declarations).....	118
Payroll_Details_To_Review [Sub].....	118
Payroll_SelectDetailsSection [Sub].....	118
Payroll_SelectReviewSection [Sub].....	118
Payroll_SelectSummarySection [Sub]	118
Payroll_Setup	120
(Declarations).....	120
Payroll_AddSelectedEmpl [Sub].....	120
Payroll_EmpToPay_SelectAll [Sub].....	120
Payroll_FilterEmployees [Sub]	120
Payroll_RemoveSelectedEmpl [Sub].....	121
Payroll_SelToPay_SelectAll [Sub].....	121
PopUp_Calendar	124
(Declarations).....	124
CalCol1 [Sub]	124
CalCol2 [Sub]	124
CalCol3 [Sub]	124
CalCol4 [Sub]	124
CalCol5 [Sub]	124
CalCol6 [Sub]	124
CalCol7 [Sub]	124
CalCol8 [Sub]	124
CalCol9 [Sub]	125
CalendarHide [Sub]	125
CalendarShow [Sub]	125
CalFormulaReplacement [Sub]	126
CheckForSheet [Sub]	126
CreateCalSht [Sub]	126
DayBtn1 [Sub]	128
DayBtn10 [Sub]	128
DayBtn11 [Sub]	129
DayBtn12 [Sub]	129
DayBtn13 [Sub]	129
DayBtn14 [Sub]	129
DayBtn15 [Sub]	129
DayBtn16 [Sub]	129
DayBtn17 [Sub]	129
DayBtn18 [Sub]	130
DayBtn19 [Sub]	130
DayBtn2 [Sub]	130
DayBtn20 [Sub]	130
DayBtn21 [Sub]	130
DayBtn22 [Sub]	130
DayBtn23 [Sub]	130
DayBtn24 [Sub]	131
DayBtn25 [Sub]	131
DayBtn26 [Sub]	131
DayBtn27 [Sub]	131
DayBtn28 [Sub]	131
DayBtn29 [Sub]	131
DayBtn3 [Sub]	131
DayBtn30 [Sub]	132
DayBtn31 [Sub]	132
DayBtn32 [Sub]	132
DayBtn33 [Sub]	132
DayBtn34 [Sub]	132
DayBtn35 [Sub]	132
DayBtn36 [Sub]	133
DayBtn37 [Sub]	133
DayBtn38 [Sub]	133
DayBtn39 [Sub]	133

DayBtn4 [Sub]	133
DayBtn40 [Sub]	133
DayBtn41 [Sub]	133
DayBtn42 [Sub]	134
DayBtn5 [Sub]	134
DayBtn6 [Sub]	134
DayBtn7 [Sub]	134
DayBtn8 [Sub]	134
DayBtn9 [Sub]	134
GroupCal [Sub]	134
MacroLinkRemover [Sub]	135
NextMonth [Sub]	136
PrevMonth [Sub]	136
ShowSettings [Sub]	136
UnGroupCal [Sub]	137
RightClick_Menu	139
(Declarations)	139
CreateLeaveEvent [Sub]	139
CreateLeaveEvent10 [Sub]	139
CreateLeaveEvent11 [Sub]	139
CreateLeaveEvent12 [Sub]	139
CreateLeaveEvent13 [Sub]	139
CreateLeaveEvent14 [Sub]	139
CreateLeaveEvent15 [Sub]	139
CreateLeaveEvent16 [Sub]	139
CreateLeaveEvent17 [Sub]	139
CreateLeaveEvent18 [Sub]	139
CreateLeaveEvent19 [Sub]	140
CreateLeaveEvent20 [Sub]	140
CreateLeaveEvent21 [Sub]	140
CreateLeaveEvent22 [Sub]	140
CreateLeaveEvent23 [Sub]	140
CreateLeaveEvent7 [Sub]	140
CreateLeaveEvent8 [Sub]	140
CreateLeaveEvent9 [Sub]	140
RightClick_CallEvents [Sub]	140
RightClick_Cancel [Sub]	141
Scheduling_Macs	143
(Declarations)	143
Sched_AddComment_Event [Sub]	143
Sched_AddComment_Holiday [Sub]	144
Sched_ClearFilters [Sub]	144
Sched_GoToEvent [Sub]	144
Sched_MonthView [Sub]	145
Sched_NextMonth [Sub]	145
Sched_NextWeek [Sub]	145
Sched_PrevMonth [Sub]	145
Sched_PrevWeek [Sub]	145
Sched_Refresh [Sub]	146
Sched_ScrollToRow [Sub]	146
Sched_ShowHideCapacity [Sub]	146
Sched_ThisMonth [Sub]	146
Sched_ThisWeek [Sub]	146
Sched_WeekView [Sub]	146
Time_Picker_Macros	148
(Declarations)	148
AMPMSel [Sub]	148
AMSel [Sub]	148
DoneBtn [Sub]	148
GrpHrs [Sub]	148
GrpMin [Sub]	148
GrpSet [Sub]	148
HideHrSel [Sub]	149
HideMinSel [Sub]	149
HoursDisplay [Sub]	149
MacroLinkRemover [Sub]	150

MinDisplay [Sub]	150
PMSEL [Sub]	151
ProtTP [Sub]	151
Select1 [Sub]	152
Select10 [Sub]	152
Select11 [Sub]	152
Select12 [Sub]	152
Select2 [Sub]	152
Select3 [Sub]	153
Select4 [Sub]	153
Select5 [Sub]	153
Select6 [Sub]	153
Select7 [Sub]	153
Select8 [Sub]	153
Select9 [Sub]	153
SelectM00 [Sub]	154
SelectM05 [Sub]	154
SelectM10 [Sub]	154
SelectM15 [Sub]	154
SelectM20 [Sub]	154
SelectM25 [Sub]	154
SelectM30 [Sub]	155
SelectM35 [Sub]	155
SelectM40 [Sub]	155
SelectM45 [Sub]	155
SelectM50 [Sub]	155
SelectM55 [Sub]	155
Set10MinInt [Sub]	155
Set15MinInt [Sub]	156
Set5MinInt [Sub]	156
SetHideShow [Sub]	156
SetThm1 [Sub]	156
SetThm2 [Sub]	156
SetThm3 [Sub]	156
SetThm4 [Sub]	156
SetThm5 [Sub]	157
TPShow [Sub]	157
UnGrpHrs [Sub]	157
UnGrpMin [Sub]	157
UnGrpSet [Sub]	158
UnProtTP [Sub]	158
TimeClock_EditSaveEntry	160
(Declarations)	160
EditTimeClock_Calculate [Sub]	160
EditTimeClock_Save [Sub]	160
TimeClock_EditEntry [Sub]	161
TimeClock_Macros	164
(Declarations)	164
Timeclock_AlreadyEntered [Sub]	164
Timeclock_DisplayPicture [Sub]	164
Timeclock_EnterID [Sub]	164
Timeclock_IDNotFound [Sub]	167
Timeclock_StartStop [Sub]	168
TimeClock_RemoteSync	170
(Declarations)	170
TimeClock_SyncRemoteEntries [Sub]	170
User_Login_Macros	173
(Declarations)	173
CheckUser [Sub]	173
CloseWorkbook [Sub]	174
HideBar [Sub]	174
LogOut [Sub]	174
ShowLoginForm [Sub]	174
StartWorkbook [Sub]	174
User_Rights_Macros	176
(Declarations)	176

Table of Contents

User_SetAs_Admin [Sub].....	176
User_SetAs_Manager [Sub].....	176
User_SetAs_TimeClock [Sub].....	176
User_SetAs_User [Sub].....	176

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:J75, 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("A9999").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("A9999").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
Else: Sheet17.Range("B" & PayItemRow).Value = "Pay Deduction" 'Item Type
109 Sheet17.Range("H" & PayItemRow).Value = Target.Row 'Save Employee Manager Row
110 Sheet17.Range("I" & PayItemRow).Value = "=Row()" 'Set Database Row
111 Range("K" & Target.Row).Value = PayItemRow
112 Else:
113     PayItemRow = Range("K" & Target.Row).Value 'On Existing Pay Item
114 End If
115 Sheet17.Cells(PayItemRow, Cells(140, Target.Column).Value).Value = Target.Value 'Tax
    Item Detail
116 End If
117 End Sub
118 Private Sub Worksheet_Deactivate()
119     SetOnkey (False)
120     Dim Cal As Shape
121     On Error Resume Next
122     Set Cal = Shapes("Calendar" )
123     On Error GoTo 0
124     If Cal Is Nothing Then 'Shape Deleted
125         Application.EnableEvents = False
126         On Error Resume Next
127         Application.Undo
128         On Error GoTo 0
129         Application.EnableEvents = True
130     End If
131 End Sub
132
133 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
134     If Target.CountLarge > 1 Then Exit Sub
135     Dim Cal As Shape 'Calendar
136     On Error Resume Next
137     Set Cal = Shapes("Calendar" )
138     On Error GoTo 0
139     If Cal Is Nothing Then 'Shape Deleted
140         Application.EnableEvents = False
141         On Error Resume Next
142         Application.Undo
143         On Error GoTo 0
144         Application.EnableEvents = True
145     End If
146     If Not Intersect(Target, Range("K16,I26,K26,I46,K46" )) Is Nothing Then 'Change this to
        any cell (or cells) you would like to have the Pop-Up Calendar Appear
147         CheckForSheet
148         CalendarShow
149     Else:
150         CheckForSheet
151         CalendarHide
152     End If
153
154
155     'Switching Horizontal Tabs in Row 4
156     If Not Intersect(Target, Range("E4:K4" )) Is Nothing Then
157         Range("B2" ).Value = Target.Column
158         Empl_SwitchHorizontalTabs
159     End If
160
161     'Switching Vertical Tabs in Column E
162     If Not Intersect(Target, Range("E66:E129" )) Is Nothing Then
163         If Target.Value = Empty Or Target.Value = "Select Option" Then Exit Sub
164         SwitchVerticalTabs
165     End If
166     If Not Intersect(Target, Range("E19:I24" )) Is Nothing Then

```

1 2

```
167     1 2 Range("B7" ).Value = Target.Row
168     If Range("J" & Target.Row).Value <> Empty Then Attach_DisplayThumbnail
169     Else: Range("B7" ).ClearContents
170     End If
171
172     'On Selection of ID Card Fields
173     If Not Intersect(Target, Range("BD7:BF24" )) Is Nothing Then EmpIDCard_FieldChange
174
175     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, 8-11
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, 10

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

E

Explicit, [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  If Not Intersect(Target, Range("F7,F8,H7,H8" )) Is Nothing Then Payroll_FilterEmployees
11  'Run Payroll Filter on Change
12 End Sub

```

```

13 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
14   If Target.CountLarge > 1 Then Exit Sub
15   Dim Cal As Shape 'Calendar
16   On Error Resume Next
17   Set Cal = Shapes("Calendar" )
18   On Error GoTo 0
19   If Cal Is Nothing Then 'Shape Deleted
20     Application.EnableEvents = False
21     On Error Resume Next
22     Application.Undo
23     On Error GoTo 0
24     Application.EnableEvents = True
25   End If
26   If Not Intersect(Target, Range("J6,J8,AM4" )) Is Nothing Then 'Change this to any cell
27   (or cells) you would like to have the Pop-Up Calendar Appear
28     CheckForSheet
29     CalendarShow
30   Else:
31     CheckForSheet
32     CalendarHide
33   End If
34   'On Selection of Employees To Add
35   If Not Intersect(Target, Range("D11:D310,L11:L310" )) Is Nothing And Cells(Target.Row,
36   Target.Column + 1).Value <> Empty Then
37     If Target.Value = "p" Then
38       Target.Value = ""
39       If Target.Column = 4 Then Range("B7" ).Value = False Else: Range("B8" ).Value =
40       False 'Select All option set to False
41     End
42   Else: Target.Value = "p"
43   End If
44   'On Selection of Employees To Review
45   If Not Intersect(Target, Range("AA9:AC310" )) Is Nothing And Range("AB" & Target.Row).
46   Value <> Empty Then
47     Range("B11" ).Value = Range("AB" & Target.Row).Value 'Set Employee To Review
48     Range("B12" ).Value = Target.Row 'Set Row To Review
49     Payroll_LoadEmpReview 'Run Employee Refresh
50   End If
51 End Sub

```

A

Application, [25](#)

C

Cal, [25](#)

CalendarHide, [25](#)

CalendarShow, [25](#)

Cells, [25](#)

CheckForSheet, [25](#)

Column, [25](#)

CountLarge, [25](#)

E

Empty, [25](#)

EnableEvents, [25](#)

Explicit, [25](#)

I

Intersect, [25](#)

P

Payroll_FilterEmployees, [25](#)

Payroll_LoadEmpReview, [25](#)

R

Range, [25](#)

Row, [25](#)

S

Shape, [25](#)

Shapes, [25](#)

Sheet4, [25](#)

T

Target, [25](#)

U

Undo, [25](#)

V

Value, [25](#)

W

Worksheet_Change, [25](#)

Worksheet_SelectionChange, [25](#)

1 Option Explicit

2

E

Explicit, [27](#)

1 Option Explicit

2

E

Explicit, [29](#)

1 Option Explicit

2

E

Explicit, [31](#)

1 Option Explicit

2

E

Explicit, [33](#)

```
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, [35](#)

E

Explicit, [35](#)

I

Intersect, [35](#)

R

Range, [35](#)

T

Target, [35](#)

V

Value, [35](#)

W

Worksheet_Change, [35](#)

Option Explicit

```
Private Sub Worksheet_Activate()
    ActiveWindow.ScrollRow = 1
End Sub
```

```
Private Sub Worksheet_Change(ByVal Target As Range)
    On Error Resume Next
```

```
    'On Change Of Custom Field Time
```

```
    If Not Intersect(Target, Range("G108:G111" )) Is Nothing Then
```

```
        Dim FieldTypeNumber As Long
```

```
        FieldTypeNumber = Application.WorksheetFunction.Match(Target.Value, Sheet4.Range(
            "FieldTypes" ), 0)
```

```
        If FieldTypeNumber <> 0 Then
```

```
            If FieldTypeNumber <> 5 Then
```

```
                Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation.Delete 'Delete
                Any possible Data validation
```

```
                Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).NumberFormat = Sheet4.Range(
                    "B" & 119 + FieldTypeNumber).Value
```

```
            Else:
```

```
                With Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation
                    .Delete
```

```
                    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween,
                    Formula1:="=CustomList" & Target.Row - 107
```

```
                    .IgnoreBlank = True
```

```
                    .InCellDropdown = True
```

```
                    .ShowInput = True
```

```
                    .ShowError = True
```

```
                End With
```

```
            End If
```

```
        End If
```

```
    End If
```

```
    'On Change Of Existing (not New) Notifications, update table below, and not on
    Notification load
```

```
    If Not Intersect(Target, Range("E507:E509,D511,G507,G508,I508" )) Is Nothing And Range(
        "B506" ).Value = False And Range("B508" ).Value = False Then
```

```
        Dim NotifRow As Long
```

```
        Dim NotifCol As Long
```

```
        NotifRow = Range("B507" ).Value 'Get Table Row
```

```
        NotifCol = Cells(Target.Row, Target.Column + 20).Value 'Get Table Column
```

```
        Cells(NotifRow, NotifCol).Value = Target.Value 'Add Change to Table Below
```

```
    End If
```

```
End Sub
```

```
Private Sub Worksheet_Deactivate()
```

```
    'Range("B3").Value = 4
```

```
    'Admin_SwitchHorizontalTabs
```

```
End Sub
```

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

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

```
    If Not Intersect(Target, Range("D4:K4" )) Is Nothing Then
```

```
        Range("B3" ).Value = Target.Column
```

```
        Range("F2" ).Select
```

```
        Admin_SwitchHorizontalTabs
```

```
    End If
```

```
    'Set Checkbox for User To Send Emails or Display Notification Pop-ups (Make changes only
```



```
1
on existing users, not blank rows
56 If Not Intersect(Target, Range("I308:I404" )) Is Nothing And Range("E" & Target.Row).
    Value <> Empty Then
57     Range("I308:I404" ).ClearContents 'Clear any Previous checkboxes
58     Target.Value = "ü"
59 End If
60 'Selection of Notifications to Load in cells above
61 If Not Intersect(Target, Range("D521:L541" )) Is Nothing And Range("D" & Target.Row).
    Value <> Empty Then
62     Range("B507" ).Value = Target.Row
63     Notif_Load
64 End If
65 End Sub
```

xlValidateList, 37

A

ActiveWindow, 37
Add, 37
Admin_SwitchHorizontalTabs, 37
AlertStyle, 37
Application, 37

C

Cells, 37
ClearContents, 38
Column, 37
CountLarge, 37

D

Delete, 37

E

Empty, 38
Explicit, 37

F

FieldTypeNumber, 37
Formula1, 37

I

IgnoreBlank, 37
InCellDropdown, 37
Intersect, 37, 38

M

Match, 37

N

Notif_Load, 38
NotifCol, 37
NotifRow, 37
NumberFormat, 37

O

Operator, 37

R

Range, 37, 38
Row, 37, 38

S

ScrollRow, 37
Sheet1, 37
Sheet4, 37
ShowError, 37
ShowInput, 37

T

Target, 37, 38

V

Validation, 37
Value, 37, 38

W

Worksheet_Activate, 37
Worksheet_Change, 37
Worksheet_Deactivate, 37
Worksheet_SelectionChange, 37
WorksheetFunction, 37

X

xlBetween, 37
xlValidAlertStop, 37

```
Option Explicit
```

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
```

```
End Sub
```

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

```
    Dim MapRng As Range
```

```
    Dim FoundMapRng As Range
```

```
    Dim CellAdd As String
```

```
    Dim SelRow As Long
```

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

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

```
    End If
```

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

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

```
    Dim EventRow As Long
```

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

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

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

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

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

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

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

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

```
    End If
```

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

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

```
End If
```

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

```
'Update Table Below
```

```
CellAdd = Target.Address
```

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

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

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

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

```
End If
```

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

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

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

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

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

```
End If
```

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

```
If Not Intersect(Target, Range("H4" )) Is Nothing And Range("B1" ).Value = False And Range
("B2" ).Value = True Then Event_AddNotifDefaults
```

```

1
47 End Sub
48
49 Private Sub Worksheet_Deactivate()
50 Dim TP As Shape
51 On Error Resume Next
52 Set TP = Shapes("TimePickGrp" )
53 On Error GoTo 0
54 If TP Is Nothing Then 'Shape Deleted
55 Application.EnableEvents = False
56 On Error Resume Next
57 Application.Undo
58 Application.EnableEvents = True
59 End If
60 End Sub
61
62 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
63 If Target.CountLarge > 1 Then Exit Sub
64 'For Pop-up Calendar
65 Dim Cal As Shape 'Calendar
66 On Error Resume Next
67 Set Cal = Shapes("Calendar" )
68 On Error GoTo 0
69 If Cal Is Nothing Then 'Shape Deleted
70 Application.EnableEvents = False
71 On Error Resume Next
72 Application.Undo
73 On Error GoTo 0
74 Application.EnableEvents = True
75 End If
76 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
77 CheckForSheet
78 CalendarShow
79 Else:
80 CheckForSheet
81 CalendarHide
82 End If
83
84 'For Pop-up Date Picker
85 Dim TP As Shape 'Time Picker
86 On Error Resume Next
87 Set TP = Shapes("TimePickGrp" )
88 On Error GoTo 0
89 If TP Is Nothing Then 'Shape Deleted
90 Application.EnableEvents = False
91 On Error Resume Next
92 Application.Undo
93 On Error GoTo 0
94 Application.EnableEvents = True
95 End If
96 If Not Intersect(Target, Range("L10,L11" )) Is Nothing Then
97 'If Sheet is Protected, add Unprotect Code here such as ActiveSheet.Unprotect
>Password"
98 TPShow
99 Else:
100 Shapes("TimePickGrp" ).Visible = msoFalse
101 End If
102
103 'On Table Selection
104 If Not Intersect(Target, Range("D27:N9999" )) Is Nothing And Range("D" & Target.Row).
Value <> Empty Then

```

1 2

```
1 2
105 Range("B10" ).Value = Target.Row 'Add in Selected Row
106 Range("B4" ).Value = Range("O" & Target.Row).Value 'Add In Event ID
107 Event_Load
108 End If
109
110 'On Recurring Selection
111 If Not Intersect(Target, Range("D10" )) Is Nothing Then
112 With ActiveCell
113 If .Value = "p Recurring" Then .Value = "o Recurring" Else: .Value = "p
    Recurring"
114 .Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
115 .Characters(Start:=2, Length:=16).Font.Name = "Calibri"
116 End With
117 Range("F11" ).Select
118 End If
119
120 'On Reminders selection
121 If Not Intersect(Target, Range("D13" )) Is Nothing Then
122 With ActiveCell
123 If .Value = "p Reminders" Then .Value = "o Reminders" Else: .Value = "p
    Reminders"
124 .Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
125 .Characters(Start:=2, Length:=15).Font.Name = "Calibri"
126 If .Value = "p Reminders" Then
127 Sheet5.Shapes("EventRemindIcons" ).Visible = msoCTrue
128 Else:
129 Sheet5.Shapes("EventRemindIcons" ).Visible = msoFalse
130 Range("L13" ).Value = "No"
131 End If
132 End With
133 Range("I13" ).Select
134 End If
135 End Sub
136
137
```

A

ActiveCell, 42
Address, 40
Application, 41

C

Cal, 41
CalendarHide, 41
CalendarShow, 41
Cancel, 40
CellAdd, 40
Cells, 40
Characters, 42
CheckForSheet, 41
Column, 40
CountLarge, 41

E

Empty, 40, 41
EnableEvents, 41
EntireRow, 40
Event_AddNotifDefaults, 40
Event_Load, 42
EventRow, 40
Explicit, 40

F

Find, 40
Font, 42
FoundMapRng, 40

H

Hidden, 40

I

InStr, 40
Intersect, 40-42
IsNumeric, 40

L

Length, 42

M

MapRng, 40
msoCTrue, 42
msoFalse, 41, 42

N

Name, 42

R

Range, 40-42
Row, 40-42

S

SelRow, 40
Shape, 41
Shapes, 41, 42
Sheet2, 40
Sheet5, 40, 42
Sheet7, 40
Start, 42

T

Target, 40-42
TP, 41
TPShow, 41

U

Undo, 41

V

Value, 40-42
Visible, 41, 42

W

Worksheet_BeforeDoubleClick, 40
Worksheet_Change, 40
Worksheet_Deactivate, 41
Worksheet_SelectionChange, 41
WrapText, 40

X

xlValues, 40
xlWhole, 40

```
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
5         <> Empty Then
6             Range("BA1" ).Value = Target.Row
7         Else: Range("BA1" ).ClearContents
8     End If
End Sub
```

C

ClearContents, [44](#)

E

Empty, [44](#)

Explicit, [44](#)

I

Intersect, [44](#)

R

Range, [44](#)

Row, [44](#)

T

Target, [44](#)

V

Value, [44](#)

W

Worksheet_SelectionChange, [44](#)

1 Option Explicit

2

E

Explicit, [46](#)

```
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, [48](#)
ActiveWindow, [48](#)
AddCol, [48](#)
Address, [48](#)
Application, [48](#)
AutoFilter, [48](#)
AutoFilterMode, [48](#)

C

Cells, [48](#)
ClearComments, [48](#)
Column, [48](#)
Count, [48](#)
Criteria1, [48](#)

D

DisplayCommentIndicator, [48](#)

E

Empty, [48](#)
Explicit, [48](#)

F

Field, [48](#)

I

Intersect, [48](#)

R

Range, [48](#)
Row, [48](#)

S

Sched_AddComment_Event, [48](#)
Sched_AddComment_Holiday, [48](#)
Sched_MonthView, [48](#)
Sched_Refresh, [48](#)
Sched_ScrollToRow, [48](#)
Sched_WeekView, [48](#)
ScrollRow, [48](#)
ShowAllData, [48](#)

T

Target, [48](#)

V

Value, [48](#)

W

Worksheet_Activate, [48](#)
Worksheet_Change, [48](#)
Worksheet_SelectionChange, [48](#)

X

xlCommentIndicatorOnly, [48](#)

1 Option Explicit

2

E

Explicit, [50](#)

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

A

ActiveSheet, [52](#)
Application, [52](#)

C

Cancel, [52](#)
CodeName, [52](#)
CommandBars, [52](#)
Controls, [52](#)

D

Delete, [52](#)

E

Explicit, [52](#)

R

Range, [52](#)
RightClick_CallEvents, [52](#)

S

SetOnkey, [52](#)
Sh, [52](#)

T

Target, [52](#)

W

Workbook_Activate, [52](#)
Workbook_BeforeClose, [52](#)
Workbook_Deactivate, [52](#)
Workbook_SheetBeforeRightClick, [52](#)


```
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, [54](#)
ClsBtn_Click, [54](#)
CreateBtn_Click, [54](#)
CreatePayPeriod, [54](#)
CreatePayPeriods, [54](#)

E

Explicit, [54](#)

H

Hide, [54](#)

```
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 Private Sub OkBtn_Click()
54     EditTimeClock_Save
55 End Sub
56
57 Private Sub UserForm_Activate()
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
1
57 ClockIn.text = Format(ClockIn.text, "h:mm AM/PM" )
58 BreakOut.text = Format(BreakOut.text, "h:mm AM/PM" )
59 BreakIn.text = Format(BreakIn.text, "h:mm AM/PM" )
60 ClockOut.text = Format(ClockOut.text, "h:mm AM/PM" )
61 EmplName.text = Sheet8.Range("A55" ).Value 'Employee Name
62 End Sub
63
64
65
```

B

BreakIn, [56](#), [57](#)
BreakIn_AfterUpdate, [56](#)
BreakInNow_MouseDown, [56](#)
BreakOut, [56](#), [57](#)
BreakOut_AfterUpdate, [56](#)
BreakOutNow_MouseDown, [56](#)
Button, [56](#)

C

CancelBtn_MouseDown, [56](#)
ClockIn, [56](#), [57](#)
ClockIn_AfterUpdate, [56](#)
ClockInNow_MouseDown, [56](#)
ClockOut, [56](#), [57](#)
ClockOut_AfterUpdate, [56](#)
ClockOutNow_MouseDown, [56](#)

E

EditTimeClock, [56](#)
EditTimeClock_Calculate, [56](#)
EditTimeClock_Save, [56](#)
EmplName, [57](#)
Explicit, [56](#)

F

Format, [56](#), [57](#)

H

Hide, [56](#)

N

Now, [56](#)

O

OkBtn_Click, [56](#)

R

Range, [57](#)

S

Sheet8, [57](#)
Shift, [56](#)

T

text, [57](#)

U

UserForm_Activate, [56](#)

V

Value, [56](#), [57](#)

X

X, [56](#)

Y

Y, [56](#)

```
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, [59](#)
CloseBtn_Click, [59](#)

E

Explicit, [59](#)

H

Hide, [59](#)
HideBar, [59](#)

L

LoginForm, [59](#)

O

OkBtn_Click, [59](#)

U

UserForm_Initialize, [59](#)

```

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, [61](#)
Application, [61](#)

C

Cells, [61](#), [62](#)
ClearContents, [61](#)

E

Empty, [61](#), [62](#)
Explicit, [61](#)

F

FileDialog, [61](#)
FileFldr, [61](#)
Filters, [61](#)

L

LastRow, [61](#)

M

MsgBox, [61](#), [62](#)
msoCTrue, [61](#), [62](#)
msoFalse, [61](#), [62](#)
msoFileDialogFilePicker, [61](#)

N

NoSelection, [61](#)
Notif_AddEmailAttach, [61](#)
Notif_AddNew, [61](#)
Notif_CancelNew, [61](#)
Notif_Delete, [61](#)
Notif_Load, [61](#)
Notif_Save, [61](#)
NotifCol, [61](#), [62](#)
NotifRow, [61](#), [62](#)

R

Range, [61](#), [62](#)
Row, [61](#), [62](#)

S

SelectedItems, [61](#)
Shapes, [61](#), [62](#)
Sheet4, [61](#), [62](#)
Show, [61](#)

T

Title, [61](#)

V

Value, [61](#), [62](#)
vbNo, [61](#)
vbYesNo, [61](#)
Visible, [61](#), [62](#)

X

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

```

1 Option Explicit
2
3 Sub BrowseEmployeeFolder()
4     Dim EmpFldr As FileDialog
5     Dim strPath As String
6     Set EmpFldr = Application.FileDialog(msoFileDialogFolderPicker)
7     With EmpFldr
8         .Title = "Select a Folder"
9         .AllowMultiSelect = False
10        If .Show <> -1 Then GoTo NoSelection
11        Sheet4.Range("EmplAttachFolder" ).Value = .SelectedItems(1)
12    End With
13    NoSelection:
14 End Sub
15 Sub BrowseRemoteTimeClockFolder()
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("RemoteTimeClockFolder" ).Value = .SelectedItems(1)
24     End With
25     NoSelection:
26 End Sub
27 Sub CreatePayPeriods()
28     Dim PayrollQty As Long
29     Dim StartRow As Long
30     Dim FormulaRow As Long
31     Dim StartDate As Date
32     Dim NextDate As Date
33     Dim Frequency As String
34     With Sheet4
35         If .Range("B4" ).Value = Empty Or .Range("B5" ).Value = Empty Or .Range("B6" ).Value =
36             Empty Then
37             MsgBox "Please make sure that all fields have been filled in before Creating Pay
38                 Periods"
39             Exit Sub
40         End If
41         CreatePayPeriod.Hide
42         Frequency = .Range("B4" ).Value 'Frequency
43         StartDate = .Range("B5" ).Value 'Start Date
44         PayrollQty = .Range("B6" ).Value 'Payroll Qty
45
46         Select Case Frequency
47         Case Is = "Weekly"
48             FormulaRow = 15
49         Case Is = "Bi-Weekly"
50             FormulaRow = 17
51         Case Is = "Semi-Monthly"
52             FormulaRow = 19
53         Case Is = "Monthly"
54             FormulaRow = 21
55         End Select
56
57         If .Range("B7" ).Value = True Then 'New List
58             .Range("D209:F304" ).ClearContents
59             StartRow = 209
60         Else:
61             StartRow = .Range("D304" ).End(xlUp).Row + 1 'First Available Row
62         End If
63     End With
64 End Sub

```

1 2 3

```
60     1 2 3 End If
61     .Range("E" & StartRow).Value = StartDate
62
63     .Range("B" & FormulaRow).Copy
64     .Range("F" & StartRow & ":F" & StartRow + PayrollQty - 1).PasteSpecial
        xlPasteFormulas
65     .Range("A" & FormulaRow + 1).Copy
66     .Range("E" & StartRow + 1 & ":E" & StartRow + PayrollQty - 1).PasteSpecial
        xlPasteFormulas
67     .Range("B14" ).Copy
68     .Range("D" & StartRow & ":D" & StartRow + PayrollQty - 1).PasteSpecial
        xlPasteFormulas 'Create Pay Period Names
69 End With
70 Application.CutCopyMode = False
71 End Sub
72
73 Sub ShowPayPeriodForm()
74     CreatePayPeriod.Show
75 End Sub
```

A

AllowMultiSelect, [64](#)
Application, [64](#), [65](#)

B

BrowseEmployeeFolder, [64](#)
BrowseRemoteTimeClockFolder, [64](#)

C

ClearContents, [64](#)
Copy, [65](#)
CreatePayPeriod, [64](#), [65](#)
CreatePayPeriods, [64](#)
CutCopyMode, [65](#)

E

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

F

FileDialog, [64](#)
FormulaRow, [64](#), [65](#)
Frequency, [64](#)

H

Hide, [64](#)

M

MsgBox, [64](#)
msoFileDialogFolderPicker, [64](#)

N

NextDate, [64](#)
NoSelection, [64](#)

P

PasteSpecial, [65](#)
PayrollQty, [64](#), [65](#)

R

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

S

SelectedItems, [64](#)
Sheet4, [64](#)
Show, [64](#), [65](#)
ShowPayPeriodForm, [65](#)
StartDate, [64](#), [65](#)
StartRow, [64](#), [65](#)
strPath, [64](#)

T

Title, [64](#)

V

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

X

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

```
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, [67](#)

E

EntireRow, [67](#)

Explicit, [67](#)

F

FirstRow, [67](#)

H

Hidden, [67](#)

M

msoCTrue, [67](#)

msoFalse, [67](#)

R

Range, [67](#)

ResetCalc, [67](#)

S

SelCol, [67](#)

Shapes, [67](#)

Sheet4, [67](#)

StopCalc, [67](#)

V

Value, [67](#)

Visible, [67](#)

```

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 > 164 And ActiveCell.Row < 184 Then GetTabOrder = Array("G182" ,
33       "I182" , "K182" )
34   End If
35   ' If ActiveCell.Row > 18 Then SetOnkey (False)   'GetTabOrder = Array("")
36 End Function
37 Sub SetOnkey(ByVal state As Boolean)
38   'Compiled By Randy Austin
39   'Workbook Provided By www.ExcelForFreelancers.com
40   If state Then
41     With Application
42       .OnKey "{TAB}" , "'TabRange xlNext'"
43       .OnKey "~" , "'TabRange xlNext'"
44       .OnKey "{ENTER}" , "'TabRange xlNext'"
45       .OnKey "{RIGHT}" , "'TabRange xlNext'"
46       .OnKey "{LEFT}" , "'TabRange xlPrevious'"
47       .OnKey "+{TAB}" , "'TabRange xlPrevious'"
48       .OnKey "{DOWN}" , "'UpOrDownArrow xlDown'"
49       .OnKey "{UP}" , "'UpOrDownArrow xlUp'"
50     End With
51   Else
52     'reset keys
53     With Application
54       .OnKey "{ENTER}"
55       .OnKey "{TAB}"
56       .OnKey "~"
57       .OnKey "+{TAB}"
58       .OnKey "{RIGHT}"
59       .OnKey "{LEFT}"
60       .OnKey "{DOWN}"

```

1 2 3


```

59     1 2 3     .OnKey "{UP}"
60         End With
61     End If
62 End Sub
63
64 Sub TabRange(Optional iDirection As Integer = xlNext)
65
66     Dim vTabOrder As Variant, m As Variant
67     Dim lItems As Long, iAdjust As Long
68     On Error GoTo ExitSub
69     '--get the tab order from shared function
70     vTabOrder = GetTabOrder
71     lItems = UBound(vTabOrder) - LBound(vTabOrder) + 1
72
73     On Error Resume Next
74     m = Application.Match(ActiveCell.Address(0, 0), vTabOrder, False)
75
76
77     '--if activecell is not in Tab Order return to the first cell
78     If IsError(m) Then
79         m = 1
80     Else
81         '--get adjustment to index
82         iAdjust = IIf(iDirection = xlPrevious, -1, 1)
83
84         '--calculate new index wrapping around list
85         m = (m + lItems + iAdjust - 1) Mod lItems + 1
86     End If
87
88     '--select cell adjusting for Option Base 0 or 1
89     Application.EnableEvents = False
90     Range(vTabOrder(m + (LBound(vTabOrder) = 0))).Select
91
92     ExitSub:
93     Application.EnableEvents = True
94
95
96 End Sub
97
98 Sub UpOrDownArrow(Optional iDirection As Integer = xlUp)
99
100     Dim vTabOrder As Variant
101     Dim lRowClosest As Long, lRowTest As Long
102     Dim i As Long, iSign As Integer
103
104     Dim sActiveCol As String
105     Dim bFound As Boolean
106
107     '--get the tab order from shared function
108     vTabOrder = GetTabOrder
109     On Error Resume Next
110     If vTabOrder = Empty Then Exit Sub
111     '--find TabCells in same column as ActiveCell in iDirection
112     '-- rTest will include ActiveCell
113
114     sActiveCol = GetColLtr(ActiveCell.Address(0, 0))
115
116     iSign = IIf(iDirection = xlDown, -1, 1)
117     lRowClosest = IIf(iDirection = xlDown, Rows.Count + 1, 0)
118
119     For i = LBound(vTabOrder) To UBound(vTabOrder)

```

1 2

```
120 1 2
121     If GetColCtr(CStr(vTabOrder(i))) = sActiveCol Then
122         lRowTest = Range(CStr(vTabOrder(i))).Row
123         '--find closest cell to ActiveCell in rTest
124     If iSign * lRowTest > iSign * lRowClosest And _
125         iSign * lRowTest < iSign * ActiveCell.Row Then
126         '--at least one cell in iDirection of same columnn
127         bFound = True
128         lRowClosest = lRowTest
129     End If
130 End If
131 Next i
132
133     If bFound Then
134         Application.EnableEvents = False
135         Cells(lRowClosest, ActiveCell.Column).Select
136         Application.EnableEvents = True
137     End If
138 End Sub
```

→, 71

A

ActiveCell, 69-71
ActiveSheet, 69
Address, 70
Application, 69-71
Array, 69

B

bFound, 70, 71

C

Cells, 71
CodeName, 69
Column, 71
Count, 70

D

do_nothing, 69

E

Empty, 70
EnableEvents, 70, 71
ExitSub, 70

G

GetColLtr, 69-71
GetTabOrder, 69, 70

I

i, 70, 71
iAdjust, 70
iDirection, 70
If, 70
iPos, 69
IsError, 70
iSign, 70, 71
IsNumeric, 69

L

LBound, 70
lItems, 70
lRowClosest, 70, 71
lRowTest, 70, 71

M

m, 70
Match, 70
Mid, 69

O

OnKey, 69, 70

R

Range, 70, 71
Row, 69, 71
Rows, 70

S

sActiveCol, 70, 71
sAddr, 69
SetOnkey, 69
state, 69
sTest, 69

T

TabRange, 70

U

UBound, 70
UpOrDownArrow, 70

V

vTabOrder, 70, 71

X

xLDown, 70
xLNext, 70
xLPrevious, 70
xLUp, 70

```
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, [73](#)

C

Calculation, [73](#)

E

EnableEvents, [73](#)

Explicit, [73](#)

R

ResetCalc, [73](#)

S

ScreenUpdating, [73](#)

StopCalc, [73](#)

X

xlCalculationAutomatic, [73](#)

xlCalculationManual, [73](#)

```

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

```



```
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, [75](#), [77](#)
ActiveCell, [76](#)
Add, [75](#), [76](#)
AdvancedFilter, [77](#)
Application, [75](#)
Attach_Add, [75](#)
Attach_Delete, [75](#)
Attach_DisplayThumbnail, [75](#), [76](#)
Attach_Filter, [77](#)
Attach_FilterClear, [77](#)
Attach_Open, [77](#)
Attach_Refresh, [75-77](#)
Attach_ShowAll, [78](#)
AttRow, [76-78](#)
AttTableRow, [76](#)

B

BadFolder, [75](#)
BrowseEmployeeFolder, [75](#)

C

ClearContents, [77](#)
CopyToRange, [77](#)
CriteriaRange, [77](#)

D

Delete, [76](#)
DestFileFolder, [75](#)
DestFilePath, [75](#)
Dir, [75](#)
DisplayAsIcon, [76](#)

E

EmpID, [75-77](#)
EmpName, [75](#)
Empty, [76](#), [77](#)
EntireRow, [76](#), [77](#)
Explicit, [75](#)

F

FileCopy, [75](#)
FileDialog, [75](#)
FileFldr, [75](#)
FileName, [75](#), [76](#)
FilePath, [76](#)
FileType, [75](#), [76](#)
Filters, [75](#)
FiltRow, [77](#), [78](#)
FollowHyperlink, [77](#)

H

Height, [76](#)
Hidden, [77](#)

I

IncrementLeft, [76](#)
IncrementTop, [76](#)
Insert, [76](#)
InStr, [75](#)

L

LastAttRow, [75](#), [77](#)
LastFiltRow, [77](#)
LCase, [76](#)
Left, [76](#)
Len, [75](#)
Link, [76](#)
LockAspectRatio, [76](#)

M

MkDir, [75](#)
MsgBox, [75-77](#)
msoCTrue, [76](#)
msoFileDialogFilePicker, [75](#)

N

Name, [76](#)
NoAttach, [76](#), [77](#)
NoData, [77](#), [78](#)
NoSelection, [75](#)
Now, [75](#)

O

OLEObjects, [76](#)
OrigFilePath, [75](#)

P

Pictures, [76](#)

R

Range, [75-78](#)
Replace, [75](#)
Right, [75](#)
Row, [75-77](#)

S

SelectedItems, [75](#)
Shapes, [76](#)
Sheet1, [75-78](#)
Sheet3, [75-78](#)
Sheet4, [75](#)
Show, [75](#)
ShowAllData, [77](#)

T

ThisWorkbook, [77](#)
ThumbType, [76](#)
Title, [75](#)
Top, [76](#)

U

unique, [77](#)
UserName, [75](#)

V

Value, [75-78](#)
vbCrLf, [75](#)
vbDirectory, [75](#)
Visible, [76](#)

W

Width, [76](#)

X

xlFilterCopy, [77](#)
xlFilterInPlace, [77](#)
xlUp, [75](#), [77](#)

```

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
102                    Picture Name
103                End With 'Shape Range
104            End With 'Pictures
105            With .Shapes("EmpIDPic" & .Cells(6, ActCol).Value) 'Add "Front" or "Back" to Picture
106                Name
107                .Left = Sheet1.Range("BH6" ).Left
108                If ActCol = 57 Then .Top = Sheet1.Range("BH6" ).Top Else: .Top = Sheet1.Range(
109                "BH16" ).Top
110                .IncrementLeft 180
111                .IncrementTop 15
112            End With
113            Exit Sub
114            NoPic:
115            End With 'End With Sheet 1
116        End Sub

```

A

ActCol, 80, 81
ActiveCell, 80
ActiveSheet, 81
ActiveWindow, 80, 81
ActRow, 80
Address, 80
AddTextbox, 80
Alignment, 80
AutoShapeType, 81

C

Cells, 80, 81
Collate, 81
Column, 80
Copies, 81

D

Delete, 81

E

EmpIDCard_CreateTextBox, 80
EmpIDCard_Customize, 80
EmpIDCard_FieldChange, 80
EmpIDCard_Hide, 80
EmpIDCard_HideCustom, 80
EmpIDCard_HidePic, 80, 81
EmpIDCard_Print, 81
EmpIDCard_Show, 81
EmpIDCard_ShowPic, 80, 81
EntireColumn, 80, 81
Explicit, 80

F

Fill, 80
Formula, 80

H

Height, 81
Hidden, 80, 81

I

IgnorePrintAreas, 81
IncrementLeft, 81
IncrementTop, 81
Insert, 81

L

Left, 81
Line, 80
LockAspectRatio, 81

M

msoAlignCenter, 80
msoCTrue, 80, 81
msoFalse, 80, 81
msoShapeRoundedRectangle, 81
msoTextOrientationHorizontal, 80
msoTrue, 81

N

Name, 80, 81
NoPic, 81

P

PageSetup, 81
ParagraphFormat, 80
PicPath, 81
Pictures, 81

PrintArea, 81
PrintOut, 81

R

Range, 80, 81
Row, 80

S

ScrollColumn, 80, 81
Selection, 80
Shape, 80
ShapeRange, 81
Shapes, 80, 81
Sheet1, 80, 81

T

text, 80
TextBox, 80
TextBoxName, 80
TextFrame2, 80
TextRange, 80
Top, 81

V

Value, 80, 81
Visible, 80, 81

```

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 = msoTrue
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
78
79   Empl_TimeclockRefresh 'Refresh Time Clock Entries
80   Empl_WeeklySchedRefresh 'Refresh Weekly Schedule
81   Empl_LeaveRefresh 'Refresh Employee Leave Items
82   Empl_EventsRefresh 'Refresh Employee Events
83   Empl_PayItemsRefresh 'Refresh Payroll Tax, Other Deductions & Other Additions
84   .Range("B1" ).Value = False 'Set Employee Load To False
85   .Range("F2" ).Select
86   End With
87   ResetCalc
88   End Sub
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
Sub Empl_PayItemsRefresh()
  Dim LastPayItemRow As Long
  Dim LastFiltRow As Long
  Dim PayItemRow As Long
  Dim EmpMgrRow As Long
  Dim ItemType As String
  With Sheet17
    .Range("M2:W9999" ).ClearContents 'Clear any Previous Criteria & Results
    .Range("M2" ).Value = Sheet1.Range("J2" ).Value 'Employee ID
    LastPayItemRow = .Range("A9999" ).End(xlUp).Row
    .Range("A2:I" & LastPayItemRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "M1:M2" ), CopyToRange:=.Range("O1:W1" ), unique:=True
    LastFiltRow = .Range("O9999" ).End(xlUp).Row
    If LastFiltRow < 2 Then GoTo NoPayItem
    For PayItemRow = 2 To LastFiltRow
      ItemType = .Range("P" & PayItemRow).Value 'Pay Item Type
      EmpMgrRow = .Range("V" & PayItemRow).Value 'Employee Manager Row
      Sheet1.Range("G" & EmpMgrRow).Value = .Range("Q" & PayItemRow).Value 'Item Name
      Sheet1.Range("K" & EmpMgrRow).Value = .Range("W" & PayItemRow).Value 'Pay Item Db
      Row
      If ItemType = "Tax Deduction" Then 'Tax Deduction Items
        Sheet1.Range("H" & EmpMgrRow).Value = .Range("R" & PayItemRow).Value 'Wage %
        Sheet1.Range("I" & EmpMgrRow).Value = .Range("S" & PayItemRow).Value 'Base
      Else: ' Pay Addition or Pay Deduction

```

```

1 2 3 4
109 Sheet1.Range("H" & EmpMgrRow).Value = .Range("T" & PayItemRow).Value 'Amount
110 Sheet1.Range("I" & EmpMgrRow).Value = .Range("U" & PayItemRow).Value
    'Frequency
111 End If
112 Next PayItemRow
113 NoPayItem:
114 End With
115 End Sub

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

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

```



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

A

Add, [83](#), [85](#)
AdvancedFilter, [83-85](#)
Application, [83](#), [85](#)
Apply, [83](#), [85](#)
Attach_Refresh, [84](#)

C

Cells, [84](#), [86](#)
Clear, [83](#), [85](#)
ClearContents, [83-85](#)
Copy, [83](#), [85](#)
CopyToRange, [83-85](#)
CriteriaRange, [83-85](#)
CutCopyMode, [83](#), [85](#)

D

DataOption, [83](#), [85](#)
Delete, [84](#)

E

EmpCol, [83](#), [84](#)
EmpID, [85](#), [86](#)
Empl_EventsRefresh, [83](#), [84](#)
Empl_LeaveRefresh, [83](#), [84](#)
Empl_Load, [83](#)
Empl_PayItemsRefresh, [84](#)
Empl_TimeclockRefresh, [84](#), [85](#)
Empl_WeeklySchedRefresh, [84](#), [85](#)
EmpMgrRow, [84](#), [85](#)
EmpRow, [83](#), [84](#)
Empty, [83](#), [85](#)
Explicit, [83](#)

F

Find, [86](#)
FoundWeeklySchedID, [85](#), [86](#)

I

ItemType, [84](#)

K

Key, [83](#), [85](#)

L

LastEventRow, [83](#)
LastFiltRow, [83-85](#)
LastLeaveRow, [83](#)
LastPayItemRow, [84](#)
LastTimeclockRow, [85](#)

M

MsgBox, [84](#)
msoCTrue, [84](#)
msoFalse, [84](#)

N

NoEvents, [83](#)
NoLeave, [83](#)
NoPayItem, [84](#), [85](#)
NoTimeclock, [85](#)
NoWeekly, [86](#)

O

Order, [83](#), [85](#)

P

PasteSpecial, [83](#), [85](#)
PayItemRow, [84](#), [85](#)

R

Range, [83-86](#)
ResetCalc, [84](#), [85](#)
Row, [83-86](#)

S

SchedRow, [85](#), [86](#)
SetRange, [83](#), [85](#)
Shapes, [84](#)
Sheet1, [83-85](#)
Sheet11, [83](#)
Sheet13, [86](#)
Sheet17, [84](#)
Sheet2, [84](#)
Sheet7, [83](#)
Sheet9, [85](#)
Show_EmplPic, [84](#)
Sort, [83](#), [85](#)
SortFields, [83](#), [85](#)
SortOn, [83](#), [85](#)
StopCalc, [84](#), [85](#)

U

unique, [83-85](#)

V

Value, [83-86](#)
Visible, [84](#)

W

WeeklyCol, [85](#), [86](#)
WeeklyRow, [85](#), [86](#)
WeeklySchedID, [85](#), [86](#)
WeeklySchedRow, [85](#)

X

xlCopy, [83](#), [85](#)
xlDescending, [83](#), [85](#)
xlFilterCopy, [83-85](#)
xlPasteFormulas, [83](#), [85](#)
xlSortNormal, [83](#), [85](#)
xlSortOnValues, [83](#), [85](#)
xlUp, [83-85](#)
xlValues, [86](#)
xlWhole, [86](#)

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:I119,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
    End With

```

```

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,
         DataOption:=xlSortNormal
116         With .Sort
117             .SetRange Range("A4:AG" & EmpRow)
118             .Apply
119         End With
120     End With
121     ResetCalc
122 End Sub
```

A

Add, 90
Application, 88, 89
Apply, 90
ArchiveRow, 88, 89

C

Cells, 89
Clear, 90
ClearContents, 88

D

DataOption, 90
Delete, 88, 89

E

EmpCol, 89
Empl_Archive, 88
Empl_CancelNew, 88
Empl_New, 88
Empl_ReActivate, 89
Empl_SaveNew, 89
Empl_SortByName, 89, 90
EmplName, 88, 89
EmplRow, 88, 89
EmpRow, 89, 90
Empty, 88, 89
EntireRow, 88, 89
Explicit, 88

F

Formula, 88, 89

K

Key, 90

M

Max, 88, 89
MsgBox, 88, 89
msoCTrue, 88, 89
msoFalse, 88, 89

O

Order, 90

R

Range, 88-90
ResetCalc, 90
Row, 88-90

S

SetRange, 90
Shapes, 88, 89
Sheet1, 88, 89
Sheet2, 88-90
Sheet6, 88, 89
Sheet8, 88, 89
Sort, 90
SortFields, 90
SortOn, 90
StopCalc, 90

V

Value, 88, 89
vbCrLf, 88
vbNo, 88, 89
vbYesNo, 88, 89
Visible, 88, 89

W

WorksheetFunction, 88, 89

X

xlAscending, 90
xlSortNormal, 90
xlSortOnValues, 90
xlUp, 88-90

```

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, [92](#)
Add_EmplPic, [92](#)
Application, [92](#)

D

DefaultPic, [92](#), [93](#)
Delete, [92](#)
DestFileFolder, [92](#)
DestFilePath, [92](#)
Dir, [92](#)

E

EmpName, [92](#)
Empty, [92](#)
Explicit, [92](#)

F

FileCopy, [92](#)
FileDialog, [92](#)
FileName, [92](#)
Filters, [92](#)

H

Height, [92](#)

I

IncrementLeft, [93](#)
IncrementTop, [93](#)
Insert, [92](#)

L

Left, [93](#)
Len, [92](#)
LockAspectRatio, [92](#)

M

MkDir, [92](#)
MsgBox, [92](#)
msoCTrue, [92](#), [93](#)
msoFalse, [92](#)
msoFileDialogFilePicker, [92](#)
msoTrue, [92](#)

N

Name, [92](#)
NoSelection, [92](#)

O

OrigFilePath, [92](#)

P

PicFile, [92](#)
PicPath, [92](#)
Pictures, [92](#)

R

Range, [92](#), [93](#)
Replace, [92](#)

S

SelectedItems, [92](#)
ShapeRange, [92](#)
Shapes, [92](#), [93](#)
Sheet1, [92](#), [93](#)
Sheet4, [92](#)
Show, [92](#)
Show_EmplPic, [92](#)

T

Title, [92](#)
Top, [93](#)

V

Value, [92](#)
vbCrLf, [92](#)
vbDirectory, [92](#)
Visible, [92](#), [93](#)

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

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, [96](#)
ActiveSheet, [95](#), [96](#)

C

CodeName, [95](#), [96](#)

E

Empl_SwitchHorizontalTabs, [95](#)
EmpTab_Events, [95](#)
EmpTab_GenInfo, [95](#)
EmpTab_Leave, [95](#)
EmpTab_PayDetails, [95](#)
EmpTab_PayHistory, [95](#)
EmpTab_Scheduling, [95](#)
EmpTab_TimeClock, [96](#)
Empty, [95](#)
EntireRow, [95](#), [96](#)
Explicit, [95](#)

F

FirstRow, [95](#), [96](#)

H

Hidden, [95](#), [96](#)

M

msoCTrue, [95](#)
msoFalse, [95](#)

R

Range, [95](#), [96](#)
ResetCalc, [95](#)
Right, [96](#)
Row, [96](#)

S

SelCol, [95](#)
SelRow, [96](#)
Shapes, [95](#)
Sheet1, [95](#), [96](#)
Show_EmplPic, [95](#)
StopCalc, [95](#)
SwitchVerticalTabs, [96](#)

V

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

```

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_CancelNew()
52     Sheet5.Range("D27" ).Select
53 End Sub
54 Sub Event_Delete()
55     If MsgBox("Are you sure you want to delete this event?" , vbYesNo, "Delete Event" ) = vbNo
56     Then Exit Sub
57     With Sheet5
58         If .Range("B3" ).Value = Empty Then Exit Sub
59         EventRow = .Range("B3" ).Value 'Event Row
60         Sheet7.Range(EventRow & ":" & EventRow).EntireRow.Delete
61     End With
62 End Sub

```

```

1 2
58     Event_Refresh 'Refresh Events
59     .Range("E27" ).Select
60     End With
61     End Sub
62
63
64     Sub Event_Load()
65         Dim SelRow As Long
66         With Sheet5
67             SelRow = .Range("B10" ).Value
68             If .Range("B3" ).Value = Empty Then
69                 MsgBox "Please select on an Event to display the Event details"
70                 Exit Sub
71             End If
72             StopCalc
73             .Range("B1" ).Value = True 'Set Event Load To True
74             EventRow = .Range("B3" ).Value
75
76             For EventCol = 1 To 26
77                 MapRng = Sheet7.Cells(1, EventCol).Value
78                 If MapRng = "D10" Or MapRng = "D13" Then 'For Recurring & Reminders
79                     If MapRng = "D10" Then
80                         If Sheet7.Cells(EventRow, EventCol).Value = "Yes" Then .Range(MapRng).Value =
81                             "p Recurring" Else: .Range(MapRng).Value = "o Recurring"
82                         .Range(MapRng).Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
83                         .Range(MapRng).Characters(Start:=2, Length:=16).Font.Name = "Calibri"
84                     End If
85                     If MapRng = "D13" Then
86                         If Sheet7.Cells(EventRow, EventCol).Value = "Yes" Then .Range(MapRng).Value =
87                             "p Reminders" Else: .Range(MapRng).Value = "o Reminders"
88                         .Range(MapRng).Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
89                         .Range(MapRng).Characters(Start:=2, Length:=15).Font.Name = "Calibri"
90                     End If
91                 Else: 'For All other fields
92                     .Range(MapRng).Value = Sheet7.Cells(EventRow, EventCol).Value 'Add Mapped Values
93                 End If
94             Next EventCol
95             If .Range("D13" ).Value = "p Reminders" Then .Shapes("EventRemindIcons" ).Visible =
96                 msoCTrue Else .Shapes("EventRemindIcons" ).Visible = msoFalse 'Show Hide Reminder
97             Icon Group
98             .Shapes("NewEventGrp" ).Visible = msoFalse
99             .Shapes("ExistEventGrp" ).Visible = msoTrue
100            .Range("B2" ).Value = False 'Set New Event To False
101            .Range("B1" ).Value = False 'Set Event Load To False
102            ResetCalc
103        End With
104    End Sub
105
106    Sub Event_Refresh()
107        Dim LastEventRow As Long
108        Dim LastFiltRow As Long
109        With Sheet5
110            StopCalc
111            'Load Events List Into Events Sheet Using Advanced Filter & Sort
112            .Range("D27:O99999" ).ClearContents 'Clear Existing Events list
113            Sheet7.Range("AD5:AQ99999" ).ClearContents 'Clear any Previous Results
114            LastEventRow = Sheet7.Range("A99999" ).End(xlUp).Row
115            DeleteFilters
116            Sheet7.Range("A4:AA" & LastEventRow).AdvancedFilter xlFilterCopy, , CopyToRange:=

```

```

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

```

A

Add, [98](#), [100](#)
AdvancedFilter, [99](#)
Application, [98](#), [100](#)
Apply, [100](#)

C

Cells, [98-100](#)
Characters, [99](#)
Clear, [100](#)
ClearContents, [98](#), [99](#)
Copy, [100](#)
CopyToRange, [99](#)
CutCopyMode, [100](#)

D

DataOption, [100](#)
Delete, [98](#)
DeleteFilters, [98](#), [99](#)

E

Empty, [98-100](#)
EntireRow, [98](#)
Event_AddEmailAttach, [98](#)
Event_AddNew, [98](#)
Event_AddNotifDefaults, [98](#)
Event_CancelNew, [98](#)
Event_Delete, [98](#)
Event_Load, [99](#)
Event_Refresh, [99](#), [100](#)
Event_SaveNew, [100](#)
EventCol, [98-100](#)
EventRow, [98-100](#)
Explicit, [98](#)

F

FileDialog, [98](#)
FileFldr, [98](#)
Filters, [98](#)
Font, [99](#)
Formula, [100](#)

I

InStr, [100](#)

K

Key, [100](#)

L

LastEventRow, [99](#)
LastFilterRow, [99](#), [100](#)
Length, [99](#)

M

MapRng, [98](#), [99](#)
MsgBox, [98-100](#)
msoCTrue, [98](#), [99](#)
msoFalse, [98-100](#)
msoFileDialogFilePicker, [98](#)
msoTrue, [99](#), [100](#)

N

Name, [99](#)
Names, [98](#)
NoEvent, [100](#)
NoSelection, [98](#)
NotifCol, [98](#)
NotifRow, [98](#)
Now, [98](#)

O

Order, [100](#)

P

PasteSpecial, [100](#)

R

Range, [98-100](#)
ResetCalc, [99](#), [100](#)
Row, [99](#), [100](#)

S

SelectedItems, [98](#)
SelRow, [99](#)
SetRange, [100](#)
Shapes, [98-100](#)
Sheet4, [98](#)
Sheet5, [98-100](#)
Sheet7, [98-100](#)
Show, [98](#)
Sort, [100](#)
SortFields, [100](#)
SortOn, [100](#)
Start, [99](#)
StopCalc, [99](#)

T

Title, [98](#)

U

unique, [100](#)

V

Value, [98-100](#)
vbNo, [98](#)
vbYesNo, [98](#)
Visible, [98-100](#)

W

WrapText, [100](#)

X

xlCopy, [100](#)
xlDescending, [100](#)
xlFilterCopy, [99](#)
xlPasteFormulas, [100](#)
xlSortNormal, [100](#)
xlSortOnValues, [100](#)
xlUp, [99](#), [100](#)


```
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, [102](#)

D

delay, [102](#)

DoEvents, [102](#)

E

EmployIDNotFound, [102](#)

Explicit, [102](#)

F

Fill, [102](#)

I

i, [102](#)

S

Shapes, [102](#)

StartTime, [102](#)

T

Timer, [102](#)

Transparency, [102](#)

V

Visible, [102](#)

```
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, [104](#)
AllowMultiSelect, [104](#)
Application, [104](#)

B

BeregnerNm, [104](#)
BeregnerRng, [104](#)

E

EmailAdd, [104](#)
Explicit, [104](#)

F

FileDialog, [104](#)
Find, [104](#)
FoundBeregner, [104](#)

G

GetFolder, [104](#)

I

InitialFileName, [104](#)

M

MailToSelected, [104](#)
msoFileDialogFolderPicker, [104](#)

N

NextCode, [104](#)

P

PDFFldr, [104](#)

R

Range, [104](#)
Row, [104](#)

S

SelectedItems, [104](#)
SelRow, [104](#)
Sheet1, [104](#)
Sheet2, [104](#)
Show, [104](#)
sItem, [104](#)
strPath, [104](#)

T

Title, [104](#)

V

Value, [104](#)

X

xlValues, [104](#)
xlWhole, [104](#)

Option Explicit

```

4 Sub Notification_Close()
5     ActiveSheet.Shapes("NotificationGrp").Delete
6 End Sub

```

```

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("A9999").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
```

—
_, 107, 108

A

ActiveCell, 108
ActiveSheet, 106, 108
Add, 106, 107
AdvancedFilter, 106
Apply, 106
Attach, 106, 107
Attachments, 107

B

BCC, 107
Body, 107

C

Characters, 108
Clear, 106
ClearContents, 106
Copy, 108
CopyToRange, 106
CreateItem, 107
CreateObject, 107
CriteriaRange, 106

D

DataOption, 106
Delete, 106
Display, 107

E

EmailBCC, 106, 107
EmailTo, 106, 107
EmplID, 106, 107
EmplIDRng, 106, 107
EmployeeEmail, 106, 107
EmployeeName, 106-108
EndOn, 106-108
EventCreatedOn, 106-108
EventName, 106-108
EventNotes, 106-108
EventRow, 106-108
EventType, 106-108
Explicit, 106

F

Find, 107
FoundEmplRng, 107

H

Height, 108

K

Key, 106

L

LastFiltRow, 106
LastRow, 106
LeaveType, 106-108
Left, 108

M

Mess, 106, 107
MsgBox, 108
msoCTrue, 108

N

NoNotif, 106

Notification_Close, 106
Notifications_Check, 106, 108
Notifications_Create, 106
NotifText, 106-108
Now, 106, 108

O

Order, 106
OutApp, 106, 107
OutMail, 106-108

P

Paste, 108

R

Range, 106-108
Replace, 107, 108
Row, 106, 107

S

SetRange, 106
Shapes, 106, 108
Sheet12, 108
Sheet14, 107
Sheet2, 107
Sheet7, 106, 107
Sort, 106
SortFields, 106
SortOn, 106
StartOn, 106-108
Subj, 106, 107
Subject, 107

T

testing, 108
text, 108
TextFrame, 108
Top, 108

U

unique, 106
UserEmail, 106, 107
UserName, 106-108

V

Value, 106-108
vbCrLf, 108
Visible, 108

W

Width, 108
WrapText, 108

X

xlAscending, 106
xlFilterCopy, 106
xlSortNormal, 106
xlSortOnValues, 106
xlUp, 106
xlValues, 107
xlWhole, 107


```

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:AM25,AP14:AQ25,AL29:AM40,AP29:AQ40").ClearContents 'clear fields
21 .Range("A014:A040,AS14:AS25,AU29:AU40").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:=xlDescending, 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
52
53 Sheet17.Range("M2:W99999").ClearContents 'Clear any Previous Criteria & Results

```

1 2 3

```

1 2 3
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         .Range("AM" & EmpPayRow - 79).Value = Sheet17.Range("R" & PayItemRow).Value
        'Wage %
66         '.Range("I" & EmpPayRow).Value = Sheet17.Range("S" & PayItemRow).Value 'Base
        will be used for calculation
67     End If
68     If ItemType = "Pay Deduction" Then
69         .Range("AP" & EmpPayRow - 94).Value = Sheet17.Range("Q" & PayItemRow).Value
        'Item Name
70         .Range("AQ" & EmpPayRow - 94).Value = Sheet17.Range("T" & PayItemRow).Value
        'Amount
71         '.Range("I" & EmpPayRow).Value = .Range("U" & PayItemRow).Value 'Frequency
        will be used for calculation
72     End If
73     If ItemType = "Pay Addition" Then
74         .Range("AL" & EmpPayRow - 99).Value = Sheet17.Range("Q" & PayItemRow).Value
        'Item Name
75         .Range("AM" & EmpPayRow - 99).Value = Sheet17.Range("T" & PayItemRow).Value
        'Amount
76         '.Range("I" & EmpPayRow).Value = .Range("U" & PayItemRow).Value 'Frequency
        will be used for calculation
77     End If
78 Next PayItemRow
79 NoPayItem:
80 'Add In Employee Leave
81 Dim LastLeaveRow As Long
82 Sheet11.Range("AA3,AB3,AA5:ADI99999" ).ClearContents 'Clear any Previous Results
83 Sheet11.Range("AA3" ).Value = EmpID 'Employee ID
84 Sheet11.Range("AB3" ).Value = "Paid" 'Only Filter Paid Leave Items
85 LastLeaveRow = Sheet11.Range("A99999" ).End(xlUp).Row
86 Sheet11.Range("A4:H" & LastLeaveRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
Sheet11.Range("AA2:AB3" ), CopyToRange:=Sheet11.Range("AA4:AE4" ), unique:=True
87 LastFiltRow = Sheet11.Range("AA99999" ).End(xlUp).Row
88 If LastFiltRow < 5 Then GoTo NoLeave
89
90 If LastFiltRow > 16 Then LastFiltRow = 16 'Set Maximum of 12 Rows
91 .Range("AP29:AP" & LastFiltRow + 24).Value = Sheet11.Range("AA5:AA" & LastFiltRow)
        .Value 'Leave Name
92 .Range("AR29:AR" & LastFiltRow + 24).Value = Sheet11.Range("AD5:AD" & LastFiltRow)
        .Value 'Allowed Leave Hours
93 NoLeave:
94
95 Else: 'Previously Saved Employee Details
96 'Run Advanced Filter on Payroll Items
97 PayrollID = .Range("B2" ).Value 'Payroll ID
98 Sheet19.Range("M3:U9999" ).ClearContents 'Clear Previous Criteria & Results
99 Sheet19.Range("M3" ).Value = PayrollID 'Set Payroll ID
100 Sheet19.Range("N3" ).Value = EmpID 'Set Employee ID
101 LastPayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row

```

1 2 3

```

1 2 3
102 If LastPayItemRow < 3 Then GoTo NoPayData
103 Sheet19.Range("A2:H" & LastPayItemRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
Sheet19.Range("M2:N3" ), CopyToRange:=Sheet19.Range("P2:U2" ), unique:=True
104 LastFiltRow = Sheet19.Range("P999" ).End(xlUp).Row
105 If LastFiltRow < 3 Then GoTo NoPayData
106 For PayItemRow = 3 To LastFiltRow
107     ItemType = Sheet19.Range("P" & PayItemRow).Value 'Set Item Type
108     PayrollRow = Sheet19.Range("T" & PayItemRow).Value 'Set Payroll row
109     If ItemType = "Regular Pay" Then .Range("AK9" ).Value = Sheet19.Range("U" &
PayItemRow).Value 'Table Row For Update
110     If ItemType = "OT Pay" Then .Range("AK10" ).Value = Sheet19.Range("U" &
PayItemRow).Value 'Table Row For Update
111     If ItemType = "Salary Pay" Then .Range("AK10" ).Value = Sheet19.Range("U" &
PayItemRow).Value 'Table Row For Update
112     If ItemType = "Tax Item" Then
113         .Range("AL" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
'Tax Item Name
114         .Range("AM" & PayrollRow).Value = Sheet19.Range("R" & PayItemRow).Value
'Tax Amount
115     End If
116     If ItemType = "Other Deduction" Then
117         .Range("AP" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
'Deduction Item Name
118         .Range("AQ" & PayrollRow).Value = Sheet19.Range("R" & PayItemRow).Value
'Deduction Amount
119     End If
120     If ItemType = "Other Addition" Then
121         .Range("AL" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
'Addition Item Name
122         .Range("AM" & PayrollRow).Value = Sheet19.Range("R" & PayItemRow).Value
'Addition Amount
123     End If
124     If ItemType = "Paid Leave" Then
125         .Range("AP" & PayrollRow).Value = Sheet19.Range("Q" & PayItemRow).Value
'Leave Item Name
126         .Range("AQ" & PayrollRow).Value = Sheet19.Range("R" & PayItemRow).Value
'Leave Hours
127     End If
128
129
130 Next PayItemRow
131
132
133
134
135
136     NoPayData:
137 End If
138
139 End With
140 ResetCalc
141 End Sub

```

A

Add, 110
AdvancedFilter, 110-112
Apply, 110

C

Clear, 110
ClearContents, 110, 111
CopyToRange, 110-112
CriteriaRange, 110-112

D

DataOption, 110

E

EmpID, 110, 111
EmpPayRow, 110, 111
EmpRow, 110
Explicit, 110

I

ItemType, 110-112

K

Key, 110

L

LastFilterRow, 110-112
LastLeaveRow, 111
LastPayItemRow, 110-112
LastTimeclockRow, 110

N

NoLeave, 111
NoPayData, 112
NoPayItem, 111
NoTimeclock, 110

O

Order, 110

P

PayItemRow, 110-112
Payroll_LoadEmpReview, 110
PayrollID, 110, 111
PayrollRow, 110, 112

R

Range, 110-112
ResetCalc, 112
ReviewRow, 110
Row, 110-112

S

SetRange, 110
Sheet11, 111
Sheet16, 110
Sheet17, 110, 111
Sheet19, 111, 112
Sheet2, 110
Sheet9, 110
Sort, 110
SortFields, 110
SortOn, 110
StopCalc, 110

U

unique, 110-112

V

Value, 110-112

X

xlDescending, 110
xlFilterCopy, 110-112
xlSortNormal, 110
xlSortOnValues, 110
xlUp, 110-112

```

1  Option Explicit
2
3  Sub Payroll_SaveEmployeePay()
4      Dim PayrollID As String
5      Dim EmpID As String
6      Dim PayType As String
7      Dim ReviewRow As Long
8      Dim PayrollRow As Long
9      Dim PayItemRow As Long
10     With Sheet16
11         'Check To Make sure Payroll has been saved, if not, save it
12
13         ReviewRow = .Range("B12").Value 'Current Employee Reivew Row
14         PayrollID = .Range("B2").Value
15         EmpID = .Range("B11").Value 'Employee ID
16         'Add In Hourly or Salary Income Items
17         PayType = .Range("AJ8").Value 'Set Pay Type
18         If PayType = "Hourly" Then 'Add in Regular & OT Amounts (If applicable)
19             If .Range("AK9").Value = Empty Then 'New Reg. Pay Item
20                 PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Row
21                 Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
22                 Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
23                 Sheet19.Range("C" & PayItemRow).Value = "Regular Pay" 'Set Pay Item Type
24                 Sheet19.Range("F" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
                PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
                locate Payroll Date
25                 Sheet19.Range("H" & PayItemRow).Value = "=Row()" 'Set Table Row
26                 .Range("AK9").Value = PayItemRow 'Set Pay Item Row
27             Else: 'Existing Regular Pay Item
28                 PayItemRow = .Range("AK9").Value 'Existing Pay Item Row
29             End If
30             Sheet19.Range("E" & PayItemRow).Value = .Range("AJ9").Value 'Set Regular Hourly
                Pay Amount
31
32             If .Range("AJ10").Value <> 0 Then 'Add in OT Amount
33                 If .Range("AK10").Value = Empty Then 'New Pay Item
34                     PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Row
35                     Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
36                     Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
37                     Sheet19.Range("C" & PayItemRow).Value = "OT Pay" 'Set Pay Item Type
38                     Sheet19.Range("F" & PayItemRow).Value =
                        "=IFERROR(INDEX(Payroll_Date,MATCH(A" & PayItemRow & ",Payroll_ID,0),1)," &
                        Chr(34) & Chr(34) & ")" 'Use formula to locate Payroll Date
39                     Sheet19.Range("H" & PayItemRow).Value = "=Row()" 'Set Table Row
40                     .Range("AK10").Value = PayItemRow 'Set Pay Item Row
41                 Else: 'Existing OT Pay Item
42                     PayItemRow = .Range("AK10").Value 'Existing Pay Item Row
43                 End If
44                 Sheet19.Range("E" & PayItemRow).Value = .Range("AJ10").Value 'Set OT Pay
                    Amount
45             End If ' End Of OT Amount Add in
46         Else: 'Salary
47             If .Range("AK10").Value = Empty Then 'New Salary Pay Item
48                 PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Row
49                 Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
50                 Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
51                 Sheet19.Range("C" & PayItemRow).Value = "Salary Pay" 'Set Pay Item Type
52                 Sheet19.Range("F" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
                PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
                locate Payroll Date
53                 Sheet19.Range("H" & PayItemRow).Value = "=Row()" 'Set Table Row

```

1 2 3 4

```

54     1 2 3 4     .Range("AK10" ).Value = PayItemRow 'Set Pay Item Row
55     } Else: 'Existing OT Pay Item
56     PayItemRow = .Range("AK10" ).Value 'Existing Pay Item Row
57     End If
58     Sheet19.Range("E" & PayItemRow).Value = .Range("AJ10" ).Value 'Set Salary Pay
59     Amount
60     End If 'End of Hourly & Salary
61     'Add In Tax Items
62     For PayrollRow = 14 To 25
63     If .Range("AL" & PayrollRow).Value = Empty Then Exit For
64     If .Range("AO" & PayrollRow).Value = Empty Then
65     PayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row + 1 'First Available Pay
66     Item 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 = "Tax Item" 'Tax Item Type
70     Sheet19.Range("F" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
71     PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
72     locate Payroll Date
73     Sheet19.Range("G" & PayItemRow).Value = PayrollRow
74     Sheet19.Range("H" & PayItemRow).Value = "=Row()" 'Set Table Row
75     .Range("AO" & PayrollRow).Value = PayItemRow
76     Else: 'Set Existing Pay Item Row
77     PayItemRow = .Range("AO" & PayrollRow).Value
78     End If
79     Sheet19.Range("D" & PayItemRow).Value = .Range("AL" & PayrollRow).Value 'Tax Item
80     Name
81     Sheet19.Range("E" & PayItemRow).Value = .Range("AM" & PayrollRow).Value 'Tax Item
82     Amount
83     Next PayrollRow
84     'Add In Other Payroll Deduction Items
85     For PayrollRow = 14 To 25
86     If .Range("AP" & PayrollRow).Value = Empty Then Exit For
87     If .Range("AS" & PayrollRow).Value = Empty Then
88     PayItemRow = Sheet19.Range("A99999" ).End(xlUp).Row + 1 'First Available Pay
89     Item Row
90     Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
91     Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
92     Sheet19.Range("C" & PayItemRow).Value = "Other Deduction" 'Set Other Deduction
93     Item Type
94     Sheet19.Range("F" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
95     PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
96     locate Payroll Date
97     Sheet19.Range("G" & PayItemRow).Value = PayrollRow
98     Sheet19.Range("H" & PayItemRow).Value = "=Row()" 'Set Table Row
99     .Range("AS" & PayrollRow).Value = PayItemRow
100    Else: 'Set Existing Pay Item Row
101    PayItemRow = .Range("AS" & PayrollRow).Value
102    End If
103    Sheet19.Range("D" & PayItemRow).Value = .Range("AP" & PayrollRow).Value 'Other
104    Deduction Name
105    Sheet19.Range("E" & PayItemRow).Value = .Range("AQ" & PayrollRow).Value 'Other
106    Deduction Amount
107    Next PayrollRow
108    'Add In Other Payroll Addition Items
109    For PayrollRow = 29 To 40
110    If .Range("AL" & PayrollRow).Value = Empty Then Exit For
111    If .Range("AO" & PayrollRow).Value = Empty Then

```

```

1 2 3 4
103 PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Pay
    Item Row
104 Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
105 Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
106 Sheet19.Range("C" & PayItemRow).Value = "Other Addition" 'Set Other Addition
    Item Type
107 Sheet19.Range("F" & PayItemRow).Value = "=IFERROR(INDEX(Payroll_Date,MATCH(A" &
    PayItemRow & ",Payroll_ID,0),1)," & Chr(34) & Chr(34) & ")" 'Use formula to
    locate Payroll Date
108 Sheet19.Range("G" & PayItemRow).Value = PayrollRow
109 Sheet19.Range("H" & PayItemRow).Value = "=Row()" 'Set Table Row
110 .Range("A0" & PayrollRow).Value = PayItemRow
111 Else: 'Set Existing Pay Item Row
112     PayItemRow = .Range("A0" & PayrollRow).Value
113 End If
114 Sheet19.Range("D" & PayItemRow).Value = .Range("AL" & PayrollRow).Value 'Other
    Deduction Name
115 Sheet19.Range("E" & PayItemRow).Value = .Range("AM" & PayrollRow).Value 'Other
    Deduction Amount
116 Next PayrollRow
117
118 'Add In Employee Paid Leave Items
119 For PayrollRow = 29 To 40
120     If .Range("AP" & PayrollRow).Value = Empty Then Exit For
121     If .Range("AQ" & PayrollRow).Value <> Empty Then 'Only add items when they contain
        hours
122         If .Range("AU" & PayrollRow).Value = Empty Then
123             PayItemRow = Sheet19.Range("A99999").End(xlUp).Row + 1 'First Available Pay
                Item Row
124             Sheet19.Range("A" & PayItemRow).Value = PayrollID 'Payroll ID
125             Sheet19.Range("B" & PayItemRow).Value = EmpID 'Employee ID
126             Sheet19.Range("C" & PayItemRow).Value = "Paid Leave" 'Set Other Addition
                Item Type
127             Sheet19.Range("G" & PayItemRow).Value = PayrollRow
128             Sheet19.Range("H" & PayItemRow).Value = "=Row()" 'Set Table Row
129             .Range("AU" & PayrollRow).Value = PayItemRow
130         Else: 'Set Existing Pay Item Row
131             PayItemRow = .Range("AU" & PayrollRow).Value
132         End If
133         Sheet19.Range("D" & PayItemRow).Value = .Range("AP" & PayrollRow).Value 'Paid
            Leave Name
134         Sheet19.Range("E" & PayItemRow).Value = .Range("AQ" & PayrollRow).Value 'Paid
            Leave Hours
135         End If 'Check for Hours To Use in AQ
136     Next PayrollRow
137     .Range("AA" & ReviewRow).Value = "ü" 'Add Completed Checkbox
138     If .Range("AB" & ReviewRow + 1).Value <> Empty Then
139         .Range("AB" & ReviewRow + 1).Select
140     Else:
141         MsgBox "You have reached the last Employee"
142     End If
143 End With
144 End Sub

```

C

Chr, [114-116](#)

E

EmpID, [114-116](#)

Empty, [114-116](#)

Explicit, [114](#)

M

MsgBox, [116](#)

P

PayItemRow, [114-116](#)

Payroll_SaveEmployeePay, [114](#)

PayrollID, [114-116](#)

PayrollRow, [114-116](#)

PayType, [114](#)

R

Range, [114-116](#)

ReviewRow, [114](#), [116](#)

Row, [114-116](#)

S

Sheet16, [114](#)

Sheet19, [114-116](#)

V

Value, [114-116](#)

X

xLUp, [114-116](#)


```
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
8                 Paid List"
9             Exit Sub
10        End If
11        LastEmpRow = .Range("M999").End(xlUp).Row 'Last Employee To Pay Row
12        .Range("AA9:AC310").ClearContents
13        .Range("AB9:AC" & LastEmpRow - 2).Value = .Range("M11:N" & LastEmpRow).Value 'Copy
14        Over Employees to be Paid
15        .Range("AA9:AA" & LastEmpRow - 2).Value = "6" 'Set Pending Status For Each Employee
16        .Range("B12").Value = 9 'Set First Employee To Review
17        .Range("B11").Value = .Range("AB9").Value 'Set First Employee ID
18        .Range("AM4").Value = Date 'Set Current Date as Default Pay Date
19        Payroll_LoadEmpReview 'Run Load Employee Review Macro for First Employee
20        Payroll_SelectReviewSection
21    End With
22 End Sub
23
24 Sub Payroll_SelectDetailsSection()
25     Dim LastEmpRow As Long
26     With Sheet16
27         .Range("C:Y").EntireColumn.Hidden = False
28         .Range("Z:BY").EntireColumn.Hidden = True
29         .Shapes("PayDetailGrp").Visible = msoTrue
30         .Shapes("PayReviewGrp").Visible = msoFalse
31         .Shapes("PaySummGrp").Visible = msoFalse
32         ActiveWindow.ScrollColumn = 3
33     End With
34 End Sub
35
36 Sub Payroll_SelectReviewSection()
37     With Sheet16
38         .Range("AZ:BY").EntireColumn.Hidden = True
39         .Range("C:Y").EntireColumn.Hidden = True
40         .Range("Z:AY").EntireColumn.Hidden = False
41         .Shapes("PayDetailGrp").Visible = msoFalse
42         .Shapes("PayReviewGrp").Visible = msoTrue
43         .Shapes("PaySummGrp").Visible = msoFalse
44         ActiveWindow.ScrollColumn = 26
45     End With
46 End Sub
47
48 Sub Payroll_SelectSummarySection()
49     With Sheet16
50         .Range("C:AY").EntireColumn.Hidden = True
51         .Range("AZ:BY").EntireColumn.Hidden = False
52         .Shapes("PayDetailGrp").Visible = msoFalse
53         .Shapes("PayReviewGrp").Visible = msoFalse
54         .Shapes("PaySummGrp").Visible = msoTrue
55     End With
56 End Sub
```

A

ActiveWindow, [118](#)

C

ClearContents, [118](#)

E

Empty, [118](#)

EntireColumn, [118](#)

Explicit, [118](#)

H

Hidden, [118](#)

L

LastEmpRow, [118](#)

M

MsgBox, [118](#)

msoFalse, [118](#)

msoTrue, [118](#)

P

Payroll_Details_To_Review, [118](#)

Payroll_LoadEmpReview, [118](#)

Payroll_SelectDetailsSection, [118](#)

Payroll_SelectReviewSection, [118](#)

Payroll_SelectSummarySection, [118](#)

R

Range, [118](#)

Row, [118](#)

S

ScrollColumn, [118](#)

Shapes, [118](#)

Sheet16, [118](#)

V

Value, [118](#)

Visible, [118](#)

X

xUp, [118](#)

```
1 Option Explicit
```

```
4 Sub Payroll_AddSelectedEmpl()
```

```
5 Dim PaidRow As Long
```

```
6 Dim SelRow As Long
```

```
7 Dim LastPaidRow As Long
```

```
8 Dim LastSelRow As Long
```

```
9 Dim FoundEmpIDRng As Range
```

```
10 With Sheet16
```

```
11 StopCalc
```

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

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

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

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

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

```
17 'Check for Duplicates
```

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

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

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

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

```
22 End If
```

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

```
24 End If
```

```
25 Next SelRow
```

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

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

```
28 With Sheet16.Sort
```

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

```
30 .Apply
```

```
31 End With
```

```
32 ResetCalc
```

```
33 End With
```

```
34 End Sub
```

```
36 Sub Payroll_EmpToPay_SelectAll()
```

```
37 Dim LastSelRow As Long
```

```
38 With Sheet16
```

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

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

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

```
42 End With
```

```
43 End Sub
```

```
45 Sub Payroll_FilterEmployees()
```

```
46 Dim LastRow As Long
```

```
47 Dim LastFiltRow As Long
```

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

```
49 With Sheet16
```

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

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

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

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

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

```
1 2
```

```

1 2
) .Value 'Employee Type Salary Frequency Criteria
55 LastRow = Sheet2.Range("A9999" ).End(xlUp).Row 'Last Employee Row
56 Sheet2.Range("A3:BC" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=Sheet2.
Range("BF2:BI3" ), CopyToRange:=Sheet2.Range("BF5:BG5" ), unique:=True
57 LastFiltRow = Sheet2.Range("BF99999" ).End(xlUp).Row
58 If LastFiltRow < 6 Then GoTo NoData
59 If .Range("B6" ).Value = True Then ' Filter By Pay Period Time Clock Data
60
61
62 End If
63 .Range("E11:F" & LastFiltRow + 5).Value = Sheet2.Range("BF6:BG" & LastFiltRow).Value
'Copy Over Employee Data
64 .Range("D11:D" & LastFiltRow + 5).Value = "p" 'Add Selected checkbox
65 NoData:
66 End With
67
68 End Sub
69
70 Sub Payroll_RemoveSelectedEmpl()
71 Dim PaidRow As Long
72 Dim SelRow As Long
73 Dim LastPaidRow As Long
74 Dim LastSelRow As Long
75 Dim FoundEmpIDRng As Range
76 With Sheet16
77 StopCalc
78 LastSelRow = .Range("D999" ).End(xlUp).Row
79 LastPaidRow = .Range("L999" ).End(xlUp).Row
80 If LastPaidRow < 10 Then LastPaidRow = 10
81 For SelRow = 11 To LastPaidRow
82 If .Range("L" & SelRow).Value = "p" Then
83 'Check for Duplicates
84 Set FoundEmpIDRng = .Range("E11:E" & LastSelRow).Find(.Range("M" & SelRow).
Value, , xlValues, xlWhole)
85 If FoundEmpIDRng Is Nothing Then 'not Found
86 LastSelRow = LastSelRow + 1
87 .Range("D" & LastSelRow & ":F" & LastSelRow).Value = .Range("L" & SelRow &
":N" & SelRow).Value 'Move Employee To First Available Row
88 End If
89 .Range("L" & SelRow & ":N" & SelRow).ClearContents 'Clear Moved Over values
90 End If
91 Next SelRow
92 .Sort.SortFields.Clear
93 .Sort.SortFields.Add Key:=.Range("N11" ), SortOn:=xlSortOnValues, Order:=xlAscending,
DataOption:=xlSortNormal
94 With Sheet16.Sort
95 .SetRange Range("L11:N" & LastPaidRow)
96 .Apply
97 End With
98 ResetCalc
99 End With
100 End Sub
101
102 Sub Payroll_SelToPay_SelectAll()
103 Dim LastSelRow As Long
104 With Sheet16
105 LastSelRow = .Range("D9999" ).End(xlUp).Row
106 If LastSelRow < 11 Then Exit Sub
107 If .Range("B7" ).Value = True Then .Range("D11:D" & LastSelRow).Value = "p" Else: .
Range("D11:D" & LastSelRow).Value = ""
108 End With

```

109

1

End Sub

A

Add, [120](#), [121](#)
AdvancedFilter, [121](#)
Apply, [120](#), [121](#)

C

Clear, [120](#), [121](#)
ClearContents, [120](#), [121](#)
CopyToRange, [121](#)
CriteriaRange, [121](#)

D

DataOption, [120](#), [121](#)

E

Explicit, [120](#)

F

Find, [120](#), [121](#)
FoundEmpIDRng, [120](#), [121](#)

K

Key, [120](#), [121](#)

L

LastFiltRow, [120](#), [121](#)
LastPaidRow, [120](#), [121](#)
LastRow, [120](#), [121](#)
LastSelRow, [120](#), [121](#)

N

NoData, [121](#)

O

Order, [120](#), [121](#)

P

PaidRow, [120](#), [121](#)
Payroll_AddSelectedEmpl, [120](#)
Payroll_EmpToPay_SelectAll, [120](#)
Payroll_FilterEmployees, [120](#)
Payroll_RemoveSelectedEmpl, [121](#)
Payroll_SelToPay_SelectAll, [121](#)

R

Range, [120](#), [121](#)
ResetCalc, [120](#), [121](#)
Row, [120](#), [121](#)

S

SelRow, [120](#), [121](#)
SetRange, [120](#), [121](#)
Sheet16, [120](#), [121](#)
Sheet2, [120](#), [121](#)
Sort, [120](#), [121](#)
SortFields, [120](#), [121](#)
SortOn, [120](#), [121](#)
StopCalc, [120](#), [121](#)

U

unique, [121](#)

V

Value, [120](#), [121](#)

X

xlAscending, [120](#), [121](#)
xlFilterCopy, [121](#)
xlSortNormal, [120](#), [121](#)

xlSortOnValues, [120](#), [121](#)

xlUp, [120](#), [121](#)

xlValues, [120](#), [121](#)

xlWhole, [120](#), [121](#)

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

1

```

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

```

1 2 3


```

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

```

1

```

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

```

1 2

```

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

```

1

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

```

1 2

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

1 2 3

```

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

```

1 2

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

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



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

1

```

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

—, 135

A

Activate, 127
 ActiveCell, 126, 128-134
 ActiveSheet, 124-137
 ActiveWorkbook, 126, 127
 ActSht, 126, 127
 Add, 126, 127
 Address, 126, 127
 After, 126
 Array, 124-126, 134-137
 AutoFill, 127, 128

B

Bold, 125, 126, 136

C

CalCol, 126, 127
 CalCol1, 124
 CalCol2, 124
 CalCol3, 124
 CalCol4, 124
 CalCol5, 124
 CalCol6, 124
 CalCol7, 124
 CalCol8, 124
 CalCol9, 125
 CalendarHide, 125
 CalendarShow, 125
 CalFormulaReplacement, 125, 126
 Cells, 126, 127
 CheckForSheet, 126
 Chr, 127, 128
 ColCnt, 126, 127
 ColNum, 126
 Copy, 126
 Count, 126
 CreateCalSht, 126
 CreateWS, 126

D

DayBtn1, 128
 DayBtn10, 128
 DayBtn11, 129
 DayBtn12, 129
 DayBtn13, 129
 DayBtn14, 129
 DayBtn15, 129
 DayBtn16, 129
 DayBtn17, 129
 DayBtn18, 130
 DayBtn19, 130
 DayBtn2, 130
 DayBtn20, 130
 DayBtn21, 130
 DayBtn22, 130
 DayBtn23, 130
 DayBtn24, 131
 DayBtn25, 131
 DayBtn26, 131
 DayBtn27, 131
 DayBtn28, 131
 DayBtn29, 131
 DayBtn3, 131
 DayBtn30, 132
 DayBtn31, 132
 DayBtn32, 132

DayBtn33, 132
 DayBtn34, 132
 DayBtn35, 132
 DayBtn36, 133
 DayBtn37, 133
 DayBtn38, 133
 DayBtn39, 133
 DayBtn4, 133
 DayBtn40, 133
 DayBtn41, 133
 DayBtn42, 134
 DayBtn5, 134
 DayBtn6, 134
 DayBtn7, 134
 DayBtn8, 134
 DayBtn9, 134
 DayCnt, 126, 127
 DayName, 124, 125, 136
 DayNum, 125, 126
 Destination, 127, 128
 DrawingObject, 125-127

E

Empty, 125, 128-134, 136

F

Fill, 124, 125, 136
 Font, 125, 126, 136
 ForeColor, 124, 125, 136
 Formula, 125-127

G

Group, 134, 135
 GroupCal, 125, 128, 134

I

InStr, 125, 135
 IsDate, 125

L

Left, 126, 135
 Len, 135

M

MacroLink, 135
 MacroLinkRemover, 125, 135
 Month, 125
 MsgBox, 126, 128
 msoCTrue, 126, 135
 msoFalse, 125, 128-134, 136
 msoTrue, 126

N

Name, 126, 127, 134, 135
 Names, 127
 NewLink, 135, 136
 NextMonth, 136
 NextShp, 135, 136
 NoCal, 125, 126, 128
 NumberFormat, 128

O

Offset, 126, 128-134
 OnAction, 125, 127, 135, 136

P

Paste, 127
 Placement, 135
 PrevMonth, 136

R

Range, 124-137
 RefersTo, 127
 RGB, 124, 125, 136
 Right, 135
 RowCnt, 126, 127
 RowNum, 126

S

SelCell, 124-126
 Selection, 124, 125, 134, 135
 Shape, 126, 135
 ShapeRange, 124, 125, 134, 135
 Shapes, 124-137
 Sheets, 124-134, 136
 ShowSettings, 136
 shp, 126, 135, 136
 Split, 135
 SplitLink, 135

T

TextFrame2, 125, 126, 136
 TextRange, 125, 126, 136
 ThisWorkbook, 126
 Top, 126

U

Ungroup, 137
 UngroupCal, 125, 127, 137

V

Value, 124-134, 136
 Visible, 125-137

W

Worksheet, 126
 ws, 126

X

xlFillDefault, 127, 128
 xlMove, 135
 xlSheetHidden, 127

Y

Year, 125

```
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 LastLeaveTypeRow As Long
99     Dim LeaveTypeRow As Long
100     Dim LeaveBtn As CommandBarPopup
101     Dim GoToEventBtn As CommandBarButton
102     Dim EditTimeClockBtn As CommandBarButton
103     With Sheet4
104         LastLeaveTypeRow = .Range("G24").End(xlUp).Row
105         On Error Resume Next
106         With Application
107             .CommandBars("Cell").Controls("Add Leave Event").Delete
108             .CommandBars("Cell").Controls("Go To Event").Delete
109             .CommandBars("Cell").Controls("Edit Time Clock Entry").Delete
110             Set LeaveBtn = .CommandBars("Cell").Controls.Add(Before:=1, Type:=msoControlPopup)
111             Set GoToEventBtn = .CommandBars("Cell").Controls.Add(Before:=2, Type:=
112                 msoControlButton)
112             Set EditTimeClockBtn = .CommandBars("Cell").Controls.Add(Before:=3, Type:=
113                 msoControlButton)
113         End With
114         With LeaveBtn
115             .Caption = "Add Leave Event"
116             For LeaveTypeRow = 7 To LastLeaveTypeRow
117                 With .Controls.Add(Type:=msoControlButton)
118                     .Caption = Sheet4.Range("G" & LeaveTypeRow).Value

```

1 2 3 4 5

```
1 2 3 4 5
119     .FaceId = 612
120     .OnAction = "CreateLeaveEvent" & LeaveTypeRow
121     End With
122     Next LeaveTypeRow
123     End With
124     With GoToEventBtn
125         .Caption = "Go To Event"
126         .FaceId = 1812
127         .OnAction = "Sched_GoToEvent"
128     End With
129     With EditTimeClockBtn
130         .Caption = "Edit Time Clock Entry"
131         .FaceId = 33
132         .OnAction = "TimeClock_EditEntry"
133     End With
134 End With
135 End Sub
136 Sub RightClick_Cancel()
137     On Error Resume Next
138     With Application
139         .CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
140         .CommandBars("Cell" ).Controls("Go To Event" ).Delete
141     End With
142     On Error GoTo 0
143 End Sub
```


A

Activate, 139
ActiveCell, 139
Add, 140
Application, 140, 141

B

Before, 140

C

Caption, 140, 141
Cells, 139
Column, 139
Columns, 139
CommandBarButton, 140
CommandBarPopup, 140
CommandBars, 140, 141
Controls, 140, 141
Count, 139
CreateLeaveEvent, 139, 140
CreateLeaveEvent10, 139
CreateLeaveEvent11, 139
CreateLeaveEvent12, 139
CreateLeaveEvent13, 139
CreateLeaveEvent14, 139
CreateLeaveEvent15, 139
CreateLeaveEvent16, 139
CreateLeaveEvent17, 139
CreateLeaveEvent18, 139
CreateLeaveEvent19, 140
CreateLeaveEvent20, 140
CreateLeaveEvent21, 140
CreateLeaveEvent22, 140
CreateLeaveEvent23, 140
CreateLeaveEvent7, 140
CreateLeaveEvent8, 140
CreateLeaveEvent9, 140

D

DateRow, 139
Delete, 140, 141

E

EditTimeClockBtn, 140, 141
EmpRow, 139
Event_AddNew, 139
Explicit, 139

F

FacelId, 141

G

GoToEventBtn, 140, 141

L

LastLeaveTypeRow, 140
LeaveBtn, 140
LeaveTypeRow, 140, 141

M

msoControlButton, 140
msoControlPopup, 140

O

OnAction, 141

R

Range, 139, 140
RightClick_CallEvents, 140

RightClick_Cancel, 141
Row, 139, 140

S

SelCol, 139
SelColumns, 139
Selection, 139
SelLeaveRow, 139, 140
SelRow, 139
Sheet2, 139
Sheet4, 139, 140
Sheet5, 139
Sheet8, 139

V

Value, 139, 140

X

xlUp, 140

Option Explicit

```

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

```

```

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

—, 143

A

ActCol, 143, 144
 Activate, 145
 ActiveCell, 143, 144
 ActiveSheet, 144
 ActiveWindow, 145, 146
 ActRow, 143, 144
 AddComment, 143, 144
 Application, 146
 AutoFilter, 144
 AutoFilterMode, 144
 AutoShapeType, 143, 144

B

BackColor, 143, 144
 Bold, 143, 144

C

Calculate, 146
 Cells, 143, 144
 Characters, 143, 144
 ClearComments, 143-146
 ColorIndex, 143, 144
 Column, 143, 144
 Comment, 143, 144
 Copy, 146
 CutCopyMode, 146

E

Empty, 143, 144
 EndOn, 143
 EntireColumn, 145, 146
 EntireRow, 146
 Event_Load, 145
 EventComment, 143
 EventCreatedOn, 143
 EventEmpl, 143
 EventID, 143-145
 EventIDRng, 143
 EventName, 143
 EventNotes, 143
 EventRecFreq, 143
 EventRecQty, 143
 EventRecur, 143
 EventRow, 143
 EventType, 143
 Explicit, 143

F

Fill, 143, 144
 Find, 143, 144
 Font, 143, 144
 ForeColor, 143, 144
 FoundEventIDRng, 143
 FoundHolDtRng, 144
 FreezePanels, 145, 146

H

Height, 143, 144
 Hidden, 145, 146
 HolDtRng, 144
 HolidayDt, 144
 HolidayName, 144
 HolidayRow, 144

L

Line, 143, 144

M

Month, 146
 MsgBox, 144
 msoCTrue, 145, 146
 msoFalse, 145, 146
 msoGradientDiagonalUp, 143, 144
 msoShapeRoundedRectangle, 143, 144
 msoTrue, 143, 144

N

Name, 143, 144
 Now, 146

O

OneColorGradient, 143, 144

P

PasteSpecial, 146

R

Range, 143-146
 ResetCalc, 145, 146
 RGB, 143, 144
 Row, 143, 144

S

Sched_AddComment_Event, 143
 Sched_AddComment_Holiday, 144
 Sched_ClearFilters, 144-146
 Sched_GoToEvent, 144
 Sched_MonthView, 145
 Sched_NextMonth, 145
 Sched_NextWeek, 145
 Sched_PrevMonth, 145
 Sched_PrevWeek, 145
 Sched_Refresh, 145, 146
 Sched_ScrollToRow, 146
 Sched_ShowHideCapacity, 146
 Sched_ThisMonth, 146
 Sched_ThisWeek, 146
 Sched_WeekView, 146
 ScrollRow, 146
 Shape, 143, 144
 Shapes, 145, 146
 Sheet4, 144
 Sheet5, 145
 Sheet7, 143
 Sheet8, 143-146
 ShowAllData, 144
 Size, 143, 144
 StartOn, 143
 StopCalc, 145, 146

T

text, 143, 144
 TextFrame, 143, 144

V

Value, 143-146
 vbCrLf, 143
 Visible, 143-146

W

Width, 143, 144

X

xlPasteFormulas, 146
 xlPasteValues, 146
 xlValues, 143, 144
 xlWhole, 143, 144

Y

Year, 145, 146

```
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 straggling 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
```

_, 151, 152

A

ActiveCell, 148, 149, 157
 ActiveSheet, 148-158
 AMPMSel, 148
 AMSel, 148
 Application, 150, 151
 Array, 148, 149, 151, 156, 157

C

Characters, 148-155, 157
 CurLeft, 149-151
 CurTop, 149-151

D

DoneBtn, 148

E

Empty, 149, 157
 ExitSub, 148, 149, 151, 152
 Explicit, 148

F

Fill, 149, 151, 155-157
 Font, 149, 151
 ForeColor, 149, 151, 155-157
 Format, 157

G

Group, 148, 149, 152
 GrpHrs, 148, 157
 GrpMin, 148, 157
 GrpRange, 148, 149, 151, 152
 GrpSet, 148, 157
 GrpShapes, 148, 149, 151, 152

H

Height, 150, 151
 HideHrSel, 149, 151-154, 157
 HideMinSel, 149, 154, 155, 157
 HoursDisplay, 149, 157
 Hr, 157

I

InStr, 150, 157
 Inter, 150, 151
 Intg10, 150, 151
 Intg15, 150, 151
 Intg5, 150, 151

L

Left, 150, 151, 157
 Len, 150

M

MacroLink, 150
 MacroLinkRemover, 150, 157
 MinDisplay, 150, 152-154, 156
 MsgBox, 157
 msoCTrue, 155-157
 msoFalse, 156, 157

N

Name, 148, 149, 152
 NewLink, 150
 NextShp, 150
 NoTP, 157

Now, 150, 151

O

Offset, 157
 OnAction, 150, 157

P

Placement, 152
 PMSel, 148, 151
 ProtTP, 148, 150-157

R

Range, 148, 149, 151, 156, 157
 RGB, 149, 151, 155-157
 Right, 150, 157

S

SelCell, 157
 Select1, 152
 Select10, 152
 Select11, 152
 Select12, 152
 Select2, 152
 Select3, 153
 Select4, 153
 Select5, 153
 Select6, 153
 Select7, 153
 Select8, 153
 Select9, 153
 Selection, 157
 SelectM00, 154, 156
 SelectM05, 154
 SelectM10, 154
 SelectM15, 154
 SelectM20, 154
 SelectM25, 154
 SelectM30, 155
 SelectM35, 155
 SelectM40, 155
 SelectM45, 155
 SelectM50, 155
 SelectM55, 155
 Set10MinInt, 155
 Set15MinInt, 156
 Set5MinInt, 156
 SetHideShow, 156
 SetThm1, 156
 SetThm2, 156
 SetThm3, 156
 SetThm4, 156
 SetThm5, 157
 Shape, 150
 Shapes, 148-158
 shp, 150
 Siz, 149-151
 Split, 150
 SplitLink, 150

T

text, 148-155, 157
 TextFrame, 148-155, 157
 TextFrame2, 149, 151
 TextRange, 149, 151
 Top, 150, 151, 157
 TPShow, 157

U

Ungroup, 157, 158
 UnGrpHrs, 157

UnGrpMin, 157, 158
 UnGrpSet, 157, 158
 UnProtTP, 148-158

V

Value, 148, 149, 157
 Visible, 148-157

W

Wait, 150, 151
 Width, 150, 151

```

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    With EditTimeClock
42        If .ClockIn.Value = Empty Then
43            MsgBox "Please make sure at least a Clock In exists for this Time Clock Entry"
44            Exit Sub
45        End If
46        If TCRow = 0 Then TCRow = Sheet9.Range("A99999" ).End(xlUp).Row + 1
47        Sheet9.Range("A" & TCRow).Value = Sheet8.Range("A59" ).Value 'Time Clock ID
48        Sheet9.Range("B" & TCRow).Value = Sheet8.Range("A53" ).Value 'Employee ID
49        Sheet9.Range("C" & TCRow).Value = Sheet8.Range("A55" ).Value 'Employee Name
50        Sheet9.Range("D" & TCRow).Value = Sheet8.Range("A57" ).Value + Format(.ClockIn.Value,
                "0.00000" )
51        If .BreakOut.Value <> Empty Then Sheet9.Range("E" & TCRow).Value = Sheet8.Range("A57"
                ).Value + Format(.BreakOut.Value, "0.00000" )
52        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

```

```

1 2
53 If .BreakIn.Value <> Empty Then Sheet9.Range("F" & TCRow).Value = Sheet8.Range("A57" )
    .Value + Format(.BreakIn.Value, "0.00000" )
54 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
55 If .ClockOut.Value <> Empty Then Sheet9.Range("G" & TCRow).Value = Sheet8.Range("A57"
).Value + Format(.ClockOut.Value, "0.00000" )
56 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
57 Sheet9.Range("H" & TCRow).Value = .Notes.Value
58 Sheet9.Range("I" & TCRow).Value = "=Row()"
59 Sheet9.Range("J" & TCRow).Value = Sheet9.Range("J" & TCRow).Value & "Admin Edit: " &
    Application.UserName & " on: " & Now
60 .Hide
61 End With
62 Sched_Refresh 'Refresh Schedule
63 End Sub
64
65 Sub TimeClock_EditEntry()
66 Dim SelRow As Long
67 Dim SelCol As Long
68 Dim DateRow As Long
69 Dim ClockInVal As Double
70 Dim ClockOutVal As Double
71 Dim BreakOutVal As Double
72 Dim BreakInVal As Double
73
74 SelRow = ActiveCell.Row
75 SelCol = ActiveCell.Column
76 If SelCol < 37 Then DateRow = 4 Else: DateRow = 5 'Set Date Row to 4 or 5 depending upon
    Month or Week view
77 With Sheet8
78 .Range("A53" ).Value = .Range("B" & SelRow).Value 'Add Employee ID
79 .Range("A57" ).Value = .Cells(DateRow, SelCol).Value 'Add Selected Date
80 If .Range("A51" ).Value <> Empty Then 'Existing Time Clock Row
81
82 TCRow = .Range("A51" ).Value
83 ClockInVal = Sheet9.Range("D" & TCRow).Value 'Clock In
84 BreakOutVal = Sheet9.Range("E" & TCRow).Value 'Break out
85 BreakInVal = Sheet9.Range("F" & TCRow).Value 'Break In
86 ClockOutVal = Sheet9.Range("G" & TCRow).Value 'Clock In
87 With EditTimeClock
88 If ClockInVal = 0 Then .ClockIn.Value = "" Else: .ClockIn.Value = ClockInVal
    'Clock In
89 If BreakOutVal = 0 Then .BreakOut.Value = "" Else: .BreakOut.Value = BreakOutVal
    'Clock Out Break
90 If BreakInVal = 0 Then .BreakIn.Value = "" Else: .BreakIn.Value = BreakInVal
    'Clock In Break
91 If ClockOutVal = 0 Then .ClockOut.Value = "" Else: .ClockOut.Value =
    ClockOutVal 'Clock Out
92 .Notes.Value = Sheet9.Range("H" & TCRow).Value 'Notes
93 End With
94 EditTimeClock_Calculate
95
96 Else ' New Time Clock Entry
97 With EditTimeClock 'Clear all form fields
98 .ClockIn.Value = vbNullString
99 .BreakOut.Value = vbNullString
100 .BreakIn.Value = vbNullString
101 .ClockOut.Value = vbNullString

```

1 2 3 4

```
102     1 2 3 4 .Notes.Value = vbNullString
103     .RegHrsValue.Caption = 0
104     .OTHrsValue.Caption = 0
105     End With
106     End If
107     EditTimeClock.Show 'Display Edit Time Clock form
108
109     End With
110
111 End Sub
```

A

ActiveCell, 161
Application, 161

B

BreakIn, 160, 161
BreakInVal, 160, 161
BreakOut, 160, 161
BreakOutVal, 160, 161

C

Caption, 160, 162
Cells, 161
ClockIn, 160, 161
ClockInVal, 160, 161
ClockOut, 160, 161
ClockOutVal, 160, 161
Column, 161

D

DateRow, 161

E

EditTimeClock, 160-162
EditTimeClock_Calculate, 160, 161
EditTimeClock_Save, 160
Empty, 160, 161
Explicit, 160

F

Format, 160, 161

H

Hide, 161

M

MsgBox, 160

N

Notes, 161, 162
Now, 161

O

OTHrsValue, 160, 162

R

Range, 160, 161
RegHrsValue, 160, 162
Row, 160, 161

S

Sched_Refresh, 161
SelCol, 161
SelRow, 161
Sheet4, 160
Sheet8, 160, 161
Sheet9, 160, 161
Show, 162

T

TCRow, 160, 161
TimeClock_EditEntry, 161

U

UserName, 161

V

Value, 160-162
vbNullString, 161, 162

X

xlUp, 160

```

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
70     'Hide Any Remaining Pop-Up Messages
71     .Shapes("IDNotFound").Visible = msoFalse
72     .Shapes("AlreadyEntered").Visible = msoFalse
73     .Shapes("TimeclockMessage").Visible = msoFalse
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
103         'Before Adding a New Row, Double Check that Employee is Not working overnight shift
104         and clocked in the night before
105         Set FoundPrevTimeClockIDRng = TimeClockIDRng.Find(TimeClockID_Prev, , xlValues,
106         xlWhole)
107         If Not FoundPrevTimeClockIDRng Is Nothing Then 'Previous ID Found
108             TimeClockRow = FoundPrevTimeClockIDRng.Row 'Previous Day TimeClock Row
109
110             'Clock Out For Break On Specific 3 Conditions (Clock Out Break Must Be Empty,
111             Admin Enabled Breaks, Under Day Threshold)
112             If .Range("E" & TimeClockRow).Value = Empty And Sheet4.Range("BreakOption").
113             Value = "Yes" And (Now - .Range("D" & TimeClockRow).Value) * 24 < DayThreshold
114             Then
115                 .Range("E" & TimeClockRow).Value = Now 'Add Clock Out From Break
116                 EntryType = "BreakOut"
117                 GoTo EntryAdded
118             End If
119
120             'Clock In From Break On Specific 4 Conditions (Clock Out For Break is Not
121             Empty, Clock In From Break is Empty, Admin Enabled Breaks, and Under Break
122             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, 164, 167
Application, 165, 167

B

BreakThreshold, 165, 166

C

ClearContents, 165, 167

D

Day, 165
DayThreshold, 165, 166
delay, 164, 167, 168
Delete, 164
DoEvents, 164, 168

E

EmpID, 164-166
EmpName, 164-167
EmpRow, 164, 165, 167
Empty, 165-167
EntryAdded, 165-167
EntryAddedAlready, 167
EntryType, 164-167
Explicit, 164

F

Fill, 164, 168
Find, 165
FoundPrevTimeClockIDRng, 165
FoundTimeClockIDRng, 164-166

H

Height, 164

I

i, 164, 167, 168
IncrementLeft, 164
IncrementTop, 164
Insert, 164

L

Left, 164
LockAspectRatio, 164

M

MessRow, 164, 167
Month, 165
msoCTrue, 167
msoFalse, 165, 167
msoSoftEdgeType4, 164
msoTrue, 164

N

Name, 164
NoPic, 164
Now, 165-168

P

PicPath, 164
Pictures, 164

R

Range, 164-168
Replace, 167
Round, 167
Row, 165, 166
RunClk, 164, 167, 168

S

ScreenUpdating, 167
ShapeRange, 164
Shapes, 164, 165, 167
Sheet12, 164, 165, 167, 168
Sheet2, 164, 167
Sheet4, 165-167
Sheet9, 165, 167
SoftEdge, 164
StartTime, 164, 167, 168

T

text, 165
Timeclock_AlreadyEntered, 164, 167
Timeclock_DisplayPicture, 164, 167
Timeclock_EnterID, 164
Timeclock_IDNotFound, 165, 167
Timeclock_StartStop, 168
TimeClockID, 164-166
TimeClockID_Prev, 164, 165
TimeClockIDRng, 164, 165
TimeClockRow, 164-167
Timer, 164, 168
TimeRow, 164, 167
TimeValue, 167, 168
Top, 164
TotalBreakHours, 165, 167
TotalHours, 165, 167
Transparency, 164, 168

V

Value, 164-167
Visible, 164, 165, 167, 168

W

Wait, 167
WorksheetFunction, 165

X

xLUp, 166
xlValues, 165
xlWhole, 165

Y

Year, 165

```
Option Explicit
```

```
Sub TimeClock_SyncRemoteEntries()
```

```
    Dim FSO As Object
```

```
    Dim TimeClockRow As Long
```

```
    Dim TimeClockFolder As String
```

```
    Dim FileName As String
```

```
    Dim TCFile As String
```

```
    Dim TCText As String
```

```
    Dim TCTextLine As String
```

```
    Dim TimeClockID As String
```

```
    Dim EmpID As String
```

```
    Dim ClockIn As String
```

```
    Dim BreakOut As String
```

```
    Dim BreakIn As String
```

```
    Dim ClockOut As String
```

```
    Dim TimeClockIDRng As Range
```

```
    Dim FoundTimeClockIDRng As Range
```

```
    Dim EmpIDRng As Range
```

```
    Dim FoundEmpIDRng As Range
```

```
    Set TimeClockIDRng = Sheet9.Range("TimeClockList_ID" )
```

```
    Set EmpIDRng = Sheet2.Range("EmployeeID" )
```

```
    With Sheet9
```

```
        If Sheet4.Range("RemoteTimeClockFolder" ).Value = Empty Then
```

```
            Sheet4.Activate
```

```
            Sheet4.Range("H4" ).Select
```

```
            MsgBox "Please make sure to select a default Dropbox (or shared) folder in which  
remote Time Clock Entries are sent"
```

```
            BrowseRemoteTimeClockFolder
```

```
            If Sheet4.Range("RemoteTimeClockFolder" ).Value = Empty Then Exit Sub 'If there is  
still no folder then exit sub, otherwise continue
```

```
        End If
```

```
        TimeClockFolder = Sheet4.Range("RemoteTimeClockFolder" ).Value 'Set Time Clock Folder
```

```
        Set FSO = CreateObject("Scripting.FileSystemObject" ) 'Set File System Object
```

```
        If Dir(TimeClockFolder & "/Time Clock Archives/" , vbDirectory) = "" Then FSO.  
createfolder (TimeClockFolder & "/Time Clock Archives/" )
```

```
        FileName = Dir(TimeClockFolder & "\*.txt" )
```

```
        Do While Len(FileName) > 0 'Start of Loop
```

```
            TCFile = TimeClockFolder & "\" & FileName
```

```
            Open TCFile For Input As #1
```

```
            Line Input #1, TCText
```

```
            Close #1
```

```
            EmpID = Mid(TCText, 13, InStr(TCText, "Clock In:" ) - 13) 'Starting Point of  
Employee ID
```

```
            ClockIn = Mid(TCText, InStr(TCText, "Clock In:" ) + 9, InStr(TCText, "Break Out:" )  
- (InStr(TCText, "Clock In:" ) + 9))
```

```
            BreakOut = Mid(TCText, InStr(TCText, "Break Out:" ) + 10, InStr(TCText, "Break In:" )  
- (InStr(TCText, "Break Out:" ) + 10))
```

```
            BreakIn = Mid(TCText, InStr(TCText, "Break In:" ) + 9, InStr(TCText, "Clock Out:" )  
- (InStr(TCText, "Break In:" ) + 9))
```

```
            ClockOut = Mid(TCText, InStr(TCText, "Clock Out:" ) + 10, (Len(TCText) - (InStr(  
TCText, "Clock Out:" ) + 10)))
```

```
            TimeClockID = Application.WorksheetFunction.text(ClockIn, "yyyy-mm-dd-" ) & EmpID
```

```
            Set FoundTimeClockIDRng = TimeClockIDRng.Find(TimeClockID, , xlValues, xlWhole)
```

```
            If Not FoundTimeClockIDRng Is Nothing Then TimeClockRow = FoundTimeClockIDRng.Row
```

```
            Else: TimeClockRow = .Range("A99999" ).End(xlUp).Row + 1 'Use Existing Found Row
```

```
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, [170](#)
Application, [170](#)

B

BreakIn, [170](#), [171](#)
BreakOut, [170](#), [171](#)
BrowseRemoteTimeClockFolder, [170](#)

C

ClockIn, [170](#), [171](#)
ClockOut, [170](#), [171](#)
Close, [170](#)
createfolder, [170](#)
CreateObject, [170](#)

D

Destination, [171](#)
Dir, [170](#), [171](#)

E

EmpID, [170](#), [171](#)
EmpIDRng, [170](#), [171](#)
Empty, [170](#), [171](#)
Explicit, [170](#)

F

FileName, [170](#), [171](#)
Find, [170](#), [171](#)
FoundEmpIDRng, [170](#), [171](#)
FoundTimeClockIDRng, [170](#)
FSO, [170](#), [171](#)

I

Input, [170](#)
InStr, [170](#)

L

Len, [170](#)
Line, [170](#)

M

Mid, [170](#)
MoveFile, [171](#)
MsgBox, [170](#)

O

Open, [170](#)

R

Range, [170](#), [171](#)
Row, [170](#), [171](#)

S

Sheet2, [170](#), [171](#)
Sheet4, [170](#)
Sheet9, [170](#)
Source, [171](#)

T

TCTFile, [170](#), [171](#)
TCText, [170](#)
TCTextLine, [170](#)
text, [170](#)
TimeClock_SyncRemoteEntries, [170](#)
TimeClockFolder, [170](#), [171](#)
TimeClockID, [170](#), [171](#)
TimeClockIDRng, [170](#)
TimeClockRow, [170](#), [171](#)

V

Value, [170](#), [171](#)
vbDirectory, [170](#)

W

WorksheetFunction, [170](#)

X

xLUp, [170](#)
xlValues, [170](#), [171](#)
xlWhole, [170](#), [171](#)

```
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 Sheet14Activate
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
```

—
_, 173

A

Activate, 174

C

Calculate, 173
Caption, 174
CheckUser, 173
ClearContents, 174
CloseWorkbook, 174

D

DrawMenuBar, 173, 174
dwNewLong, 173

E

Empty, 173
Explicit, 173

F

FindWindow, 173, 174
frm, 174

G

GetWindowLong, 173, 174

H

Hide, 174
HideBar, 174
hWnd, 173
hWndForm, 174

L

LoginForm, 174
LogOut, 174
lpClassName, 173
lpWindowName, 173

M

Menu, 174
MsgBox, 173

N

Name, 174
nIndex, 173

P

PtrSafe, 173

R

Range, 173, 174

S

Save, 174
SetWindowLong, 173, 174
Sheet14, 173, 174
Sheet14Activate, 174
Show, 174
ShowLoginForm, 174
StartWorkbook, 174
Style, 174

T

ThisWorkbook, 174

U

User_SetAs_Admin, 174

User_SetAs_Manager, 174
User_SetAs_TimeClock, 174
User_SetAs_User, 174
UserType, 173, 174

V

Value, 173, 174
VBA7, 173
Visible, 174

W

WkSht, 174
Worksheet, 174
Worksheets, 174

X

xlSheetVeryHidden, 174
xlSheetVisible, 174

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, [176](#)

E

Explicit, [176](#)

N

Name, [176](#)

R

Range, [176](#)

S

Sheet1, [176](#)
Sheet11, [176](#), [177](#)
Sheet12, [176](#), [177](#)
Sheet13, [176](#), [177](#)
Sheet14, [176](#), [177](#)
Sheet15, [176](#), [177](#)
Sheet16, [176](#), [177](#)
Sheet2, [176](#)
Sheet3, [176](#)
Sheet4, [176](#)
Sheet5, [176](#)
Sheet6, [176](#), [177](#)
Sheet7, [176](#), [177](#)
Sheet8, [176](#), [177](#)
Sheet9, [176](#), [177](#)

T

ThisWorkbook, [176](#)

U

User_SetAs_Admin, [176](#)
User_SetAs_Manager, [176](#)
User_SetAs_TimeClock, [176](#)
User_SetAs_User, [176](#)

V

Visible, [176](#), [177](#)

W

WkSht, [176](#)
Worksheet, [176](#)
Worksheets, [176](#)

X

xlSheetHidden, [176](#), [177](#)
xlSheetVeryHidden, [176](#), [177](#)
xlSheetVisible, [176](#), [177](#)

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