

VBA SOURCE CODE BOOK



**How To Automatically Sync
Mobile Form Data Into Excel
[Employee Manager 12]**



**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



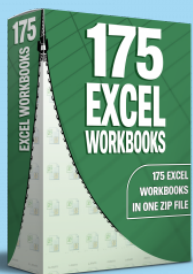
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



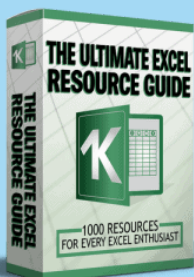
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[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.....	6
VBAProject	6
Documents	6
Sheet1.....	6
(Declarations).....	6
Worksheet_Activate [Sub].....	6
Worksheet_Change [Sub].....	6
Worksheet_Deactivate [Sub]	7
Worksheet_SelectionChange [Sub]	7
Sheet10.....	10
(Declarations).....	10
Sheet11.....	12
(Declarations).....	12
Sheet12.....	14
Worksheet_Change [Sub]	14
Sheet2.....	16
(Declarations).....	16
Sheet3.....	18
(Declarations).....	18
Worksheet_Change [Sub]	18
Sheet4.....	20
(Declarations).....	20
Worksheet_Activate [Sub].....	20
Worksheet_Change [Sub]	20
Worksheet_Deactivate [Sub].....	20
Worksheet_SelectionChange [Sub]	20
Sheet5.....	22
(Declarations).....	22
Worksheet_BeforeDoubleClick [Sub]	22
Worksheet_Change [Sub]	22
Worksheet_Deactivate [Sub].....	22
Worksheet_SelectionChange [Sub]	23
Sheet6.....	26
(Declarations).....	26
Sheet7.....	28
(Declarations).....	28
Sheet8.....	30
(Declarations).....	30
Worksheet_Activate [Sub].....	30
Worksheet_Change [Sub]	30
Worksheet_SelectionChange [Sub]	30
Sheet9.....	32
(Declarations).....	32
ThisWorkbook	34
(Declarations).....	34
Workbook_Activate [Sub].....	34
Workbook_Deactivate [Sub].....	34
Workbook_SheetBeforeRightClick [Sub].....	34
Forms	36
CreatePayPeriod	36
(Declarations).....	36
CancelBtn_Click [Sub]	36
ClsBtn_Click [Sub]	36
CreateBtn_Click [Sub].....	36
LoginForm	38
(Declarations).....	38
ToggleButton1_Click [Sub].....	38
Modules.....	40
Admin_Macros	40
(Declarations).....	40
BrowseEmployeeFolder [Sub]	40
BrowseRemoteTimeClockFolder [Sub].....	40
CreatePayPeriods [Sub].....	40
ShowPayPeriodForm [Sub]	41
Admin_Tabs	43
(Declarations).....	43

Admin_SwitchHorizontalTabs [Sub]	43
AutoTab_Macros	45
do_nothing [Sub]	45
GetColLtr [Function]	45
GetTabOrder [Function]	45
SetOnkey [Sub]	45
TabRange [Sub]	46
UpOrDownArrow [Sub]	46
CodeReset_Macs	49
(Declarations)	49
ResetCalc [Sub]	49
StopCalc [Sub]	49
Empl_Attach	51
(Declarations)	51
Attach_Add [Sub]	51
Attach_Delete [Sub]	51
Attach_DisplayThumbnail [Sub]	52
Attach_Filter [Sub]	53
Attach_FilterClear [Sub]	53
Attach_Open [Sub]	53
Attach_Refresh [Sub]	53
Attach_ShowAll [Sub]	54
Empl_IDCard	56
(Declarations)	56
EmplIDCard_CreateTextBox [Sub]	56
EmplIDCard_Customize [Sub]	56
EmplIDCard_FieldChange [Sub]	56
EmplIDCard_Hide [Sub]	56
EmplIDCard_HideCustom [Sub]	56
EmplIDCard_HidePic [Sub]	57
EmplIDCard_Print [Sub]	57
EmplIDCard_Show [Sub]	57
EmplIDCard_ShowPic [Sub]	57
Empl_Misc	59
(Declarations)	59
Emp_SaveNew [Sub]	59
Empl_Archive [Sub]	59
Empl_CancelNew [Sub]	59
Empl_EventsRefresh [Sub]	60
Empl_LeaveRefresh [Sub]	60
Empl_Load [Sub]	60
Empl_New [Sub]	61
Empl_SortByName [Sub]	62
Empl_TimeclockRefresh [Sub]	62
Empl_PicsMacs	64
(Declarations)	64
Add_EmplPic [Sub]	64
Show_EmplPic [Sub]	64
Empl_Tabs	67
(Declarations)	67
Empl_SwitchHorizontalTabs [Sub]	67
EmpTab_Events [Sub]	67
EmpTab_GenInfo [Sub]	67
EmpTab_Leave [Sub]	67
EmpTab_PayDetails [Sub]	67
EmpTab_PayHistory [Sub]	67
EmpTab_Scheduling [Sub]	67
EmpTab_TimeClock [Sub]	68
SwitchVerticalTabs [Sub]	68
Events_Misc	70
(Declarations)	70
DeleteFilters [Sub]	70
Event_AddEmailAttach [Sub]	70
Event_AddNew [Sub]	70
Event_CancelNew [Sub]	70
Event_Delete [Sub]	70

Event_Load [Sub]	70
Event_Refresh [Sub]	71
Event_SaveNew [Sub]	72
FadeOutMessage	74
(Declarations).....	74
EmployIDNotFound [Sub]	74
FolderPickMac	76
(Declarations).....	76
GetFolder [Sub]	76
MailToSelected [Sub]	76
PopUp_Calendar	78
(Declarations).....	78
CalCol1 [Sub]	78
CalCol2 [Sub]	78
CalCol3 [Sub]	78
CalCol4 [Sub]	78
CalCol5 [Sub]	78
CalCol6 [Sub]	78
CalCol7 [Sub]	78
CalCol8 [Sub]	78
CalCol9 [Sub]	79
CalendarHide [Sub]	79
CalendarShow [Sub]	79
CalFormulaReplacement [Sub]	80
CheckForSheet [Sub]	80
CreateCalSht [Sub]	80
DayBtn1 [Sub]	82
DayBtn10 [Sub]	82
DayBtn11 [Sub]	83
DayBtn12 [Sub]	83
DayBtn13 [Sub]	83
DayBtn14 [Sub]	83
DayBtn15 [Sub]	83
DayBtn16 [Sub]	83
DayBtn17 [Sub]	83
DayBtn18 [Sub]	84
DayBtn19 [Sub]	84
DayBtn2 [Sub]	84
DayBtn20 [Sub]	84
DayBtn21 [Sub]	84
DayBtn22 [Sub]	84
DayBtn23 [Sub]	84
DayBtn24 [Sub]	85
DayBtn25 [Sub]	85
DayBtn26 [Sub]	85
DayBtn27 [Sub]	85
DayBtn28 [Sub]	85
DayBtn29 [Sub]	85
DayBtn3 [Sub]	85
DayBtn30 [Sub]	86
DayBtn31 [Sub]	86
DayBtn32 [Sub]	86
DayBtn33 [Sub]	86
DayBtn34 [Sub]	86
DayBtn35 [Sub]	86
DayBtn36 [Sub]	86
DayBtn37 [Sub]	87
DayBtn38 [Sub]	87
DayBtn39 [Sub]	87
DayBtn4 [Sub]	87
DayBtn40 [Sub]	87
DayBtn41 [Sub]	87
DayBtn42 [Sub]	88
DayBtn5 [Sub]	88
DayBtn6 [Sub]	88
DayBtn7 [Sub]	88

DayBtn8 [Sub]	88
DayBtn9 [Sub]	88
GroupCal [Sub]	88
MacroLinkRemover [Sub]	89
NextMonth [Sub]	90
PrevMonth [Sub]	90
ShowSettings [Sub]	90
UnGroupCal [Sub]	91
RightClick_Menu	93
(Declarations)	93
CreateLeaveEvent [Sub]	93
CreateLeaveEvent10 [Sub]	93
CreateLeaveEvent11 [Sub]	93
CreateLeaveEvent12 [Sub]	93
CreateLeaveEvent13 [Sub]	93
CreateLeaveEvent14 [Sub]	93
CreateLeaveEvent15 [Sub]	93
CreateLeaveEvent16 [Sub]	93
CreateLeaveEvent17 [Sub]	93
CreateLeaveEvent18 [Sub]	93
CreateLeaveEvent19 [Sub]	94
CreateLeaveEvent20 [Sub]	94
CreateLeaveEvent21 [Sub]	94
CreateLeaveEvent22 [Sub]	94
CreateLeaveEvent23 [Sub]	94
CreateLeaveEvent7 [Sub]	94
CreateLeaveEvent8 [Sub]	94
CreateLeaveEvent9 [Sub]	94
RightClick_CallEvents [Sub]	94
RightClick_Cancel [Sub]	95
Scheduling_Macs	97
(Declarations)	97
Sched_AddComment_Event [Sub]	97
Sched_AddComment_Holiday [Sub]	98
Sched_ClearFilters [Sub]	98
Sched_GoToEvent [Sub]	98
Sched_MonthView [Sub]	99
Sched_NextMonth [Sub]	99
Sched_NextWeek [Sub]	99
Sched_PrevMonth [Sub]	99
Sched_PrevWeek [Sub]	99
Sched_Refresh [Sub]	100
Sched_ScrollToRow [Sub]	100
Sched_ShowHideCapacity [Sub]	100
Sched_ThisMonth [Sub]	100
Sched_ThisWeek [Sub]	100
Sched_WeekView [Sub]	100
Time_Picker_Macros	102
(Declarations)	102
AMPMSel [Sub]	102
AMSel [Sub]	102
DoneBtn [Sub]	102
GrpHrs [Sub]	102
GrpMin [Sub]	102
GrpSet [Sub]	102
HideHrSel [Sub]	103
HideMinSel [Sub]	103
HoursDisplay [Sub]	103
MacroLinkRemover [Sub]	104
MinDisplay [Sub]	104
PMSel [Sub]	105
ProtTP [Sub]	105
Select1 [Sub]	106
Select10 [Sub]	106
Select11 [Sub]	106
Select12 [Sub]	106

Select2 [Sub].....	106
Select3 [Sub].....	107
Select4 [Sub].....	107
Select5 [Sub].....	107
Select6 [Sub].....	107
Select7 [Sub].....	107
Select8 [Sub].....	107
Select9 [Sub].....	107
SelectM00 [Sub].....	108
SelectM05 [Sub].....	108
SelectM10 [Sub].....	108
SelectM15 [Sub].....	108
SelectM20 [Sub].....	108
SelectM25 [Sub].....	108
SelectM30 [Sub].....	109
SelectM35 [Sub].....	109
SelectM40 [Sub].....	109
SelectM45 [Sub].....	109
SelectM50 [Sub].....	109
SelectM55 [Sub].....	109
Set10MinInt [Sub]	109
Set15MinInt [Sub].....	110
Set5MinInt [Sub].....	110
SetHideShow [Sub]	110
SetThm1 [Sub]	110
SetThm2 [Sub]	110
SetThm3 [Sub]	110
SetThm4 [Sub]	110
SetThm5 [Sub].....	111
TPShow [Sub].....	111
UnGrpHrs [Sub]	111
UnGrpMin [Sub].....	111
UnGrpSet [Sub].....	112
UnProtTP [Sub].....	112
TimeClock_Macros.....	114
(Declarations).....	114
Timeclock_AlreadyEntered [Sub]	114
Timeclock_DisplayPicture [Sub]	114
Timeclock_EnterID [Sub].....	114
Timeclock_IDNotFound [Sub].....	117
Timeclock_StartStop [Sub].....	118
TimeClock_RemoteSync.....	120
(Declarations).....	120
TimeClock_SyncRemoteEntries [Sub].....	120

Option Explicit

```
Private Sub Worksheet_Activate()
    SetOnkey (True)
End Sub
```

```
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim EmpRow As Long
    Dim PayPeriod As Range
    Dim FoundPayPeriod As Range
    'On Employee Name Change
    If Not Intersect(Target, Range("F2" )) Is Nothing Then
        If Range("F2" ).Value <> Empty And Range("B5" ).Value <> Empty Then Empl_Load
    End If

    'Employee ID Change
    If Not Intersect(Target, Range("J2" )) Is Nothing And Range("B1" ).Value = False Then 'On
change of Employee ID but NOT on Employee Load
        If Range("J2" ).Value <> Empty And Range("B4" ).Value <> Empty Then 'Employee ID Found

            Range("F2" ).Value = Sheet2.Range("BC" & Range("B4" ).Value).Value
        Else: 'Employee ID Not Found, Return to original
            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
        End If
    End If
```

```
'On Employee Information Change
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
    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
    EmpRow = Range("B4" ).Value 'Employee Row
    Sheet2.Cells(EmpRow, Cells(Target.Row, Target.Column + 22).Value).Value = Target.Value
End If
```

```
'On Date Range History Change
If Not Intersect(Target, Range("F26, F46" )) Is Nothing And Range("F" & Target.Row).Value
<> Empty Then
    Set PayPeriod = Sheet4.Range("PayPeriod" )
    Sheet4.Range("209:304" ).EntireRow.Hidden = False
    Set FoundPayPeriod = PayPeriod.Find(Target.Value, , xlValues, xlWhole)
    If Not FoundPayPeriod Is Nothing Then
        Target.Offset(0, 2).Value = Sheet4.Range("E" & FoundPayPeriod.Row).Value 'From
        Date
        Target.Offset(0, 4).Value = Sheet4.Range("F" & FoundPayPeriod.Row).Value 'To Date
    End If
    Sheet4.Range("209:304" ).EntireRow.Hidden = True
End If
```

```
'On Employee Leave Change, Update Database Table, but not on Employee Load
If Not Intersect(Target, Range("E168:H179" )) Is Nothing And Range("B1" ).Value = False
Then
    Dim LeaveRow As Long
    Dim LeaveID As Long
    Dim LeaveListID As Range
    Set LeaveListID = Sheet11.Range("Leavelist_ID" )
    If Range("L" & Target.Row).Value = Empty Then 'New Employee Leave Item
        LeaveRow = Sheet11.Range("A99999" ).End(xlUp).Row + 1
        LeaveID = Application.WorksheetFunction.Max(LeaveListID) + 1
```

1 2 3

```

1 2 3
54 Range("L" & Target.Row).Value = LeaveRow
55 Sheet11.Range("A" & LeaveRow).Value = LeaveID
56 Sheet11.Range("B" & LeaveRow).Value = Range("J2").Value 'Employee ID
57 Sheet11.Range("C" & LeaveRow).Value = Range("F2").Value 'Employee Name
58 Sheet11.Range("H" & LeaveRow).Value = "=Row()" 'Set Row
59 Else:
60 LeaveRow = Range("L" & Target.Row).Value
61 End If
62 Sheet11.Cells(LeaveRow, Target.Column - 1).Value = Target.Value
63 End If
64 End Sub
65 Private Sub Worksheet_Deactivate()
66 SetOnkey (False)
67 Dim Cal As Shape
68 On Error Resume Next
69 Set Cal = Shapes("Calendar")
70 On Error GoTo 0
71 If Cal Is Nothing Then 'Shape Deleted
72 Application.EnableEvents = False
73 On Error Resume Next
74 Application.Undo
75 On Error GoTo 0
76 Application.EnableEvents = True
77 End If
78 End Sub
79
80 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
81 If Target.Count > 1 Then Exit Sub
82 Dim Cal As Shape 'Calendar
83 On Error Resume Next
84 Set Cal = Shapes("Calendar")
85 On Error GoTo 0
86 If Cal Is Nothing Then 'Shape Deleted
87 Application.EnableEvents = False
88 On Error Resume Next
89 Application.Undo
90 On Error GoTo 0
91 Application.EnableEvents = True
92 End If
93 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
94 CheckForSheet
95 CalendarShow
96 Else:
97 CheckForSheet
98 CalendarHide
99 End If
100
101
102 'Switching Horizontal Tabs in Row 4
103 If Not Intersect(Target, Range("E4:K4")) Is Nothing Then
104 Range("B2").Value = Target.Column
105 Empl_SwitchHorizontalTabs
106 End If
107
108 'Switching Vertical Tabs in Column E
109 If Not Intersect(Target, Range("E66:E129")) Is Nothing Then
110 If Target.Value = Empty Or Target.Value = "Select Option" Then Exit Sub
111 SwitchVerticalTabs
112 End If
113 If Not Intersect(Target, Range("E19:I24")) Is Nothing Then

```

1 2

```
114     1 2 Range("B7" ).Value = Target.Row
115     If Range("J" & Target.Row).Value <> Empty Then Attach_DisplayThumbnail
116     Else: Range("B7" ).ClearContents
117     End If
118
119     'On Selection of ID Card Fields
120     If Not Intersect(Target, Range("BD7:BF24" )) Is Nothing Then EmpIDCard_FieldChange
121
122     End Sub
```

A

Application, 6, 7
Attach_DisplayThumbnail, 8

C

Cal, 7
CalendarHide, 7
CalendarShow, 7
Cells, 6, 7
CheckForSheet, 7
ClearContents, 8
Column, 6, 7
Count, 7

E

EmpIDCard_FieldChange, 8
Empl_Load, 6
Empl_SwitchHorizontalTabs, 7
EmployIDNotFound, 6
EmpRow, 6
Empty, 6-8
EnableEvents, 7
EntireRow, 6
Explicit, 6

F

Find, 6
FoundPayPeriod, 6

H

Hidden, 6

I

Intersect, 6-8
IsNumeric, 6

L

LeaveID, 6, 7
LeaveListID, 6
LeaveRow, 6, 7

M

Max, 6

O

Offset, 6

P

PayPeriod, 6

R

Range, 6-8
Row, 6-8

S

SetOnkey, 6, 7
Shape, 7
Shapes, 7
Sheet11, 6, 7
Sheet2, 6
Sheet4, 6
SwitchVerticalTabs, 7

T

Target, 6-8

U

Undo, 7

V

Value, 6-8

W

Worksheet_Activate, 6
Worksheet_Change, 6
Worksheet_Deactivate, 7
Worksheet_SelectionChange, 7
WorksheetFunction, 6

X

xLUp, 6
xlValues, 6
xlWhole, 6

1 Option Explicit

2

E

Explicit, [10](#)

1 Option Explicit

2

E

Explicit, [12](#)


```
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, [14](#)

I

Intersect, [14](#)

R

Range, [14](#)

T

Target, [14](#)

Timeclock_EnterID, [14](#)

V

Value, [14](#)

W

Worksheet_Change, [14](#)

1 Option Explicit

2

E

Explicit, [16](#)

```
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, 18

E

Explicit, 18

I

Intersect, 18

R

Range, 18

T

Target, 18

V

Value, 18

W

Worksheet_Change, 18

```
1 Option Explicit
2
3 Private Sub Worksheet_Activate()
4     ActiveWindow.ScrollRow = 1
5 End Sub
6
7 Private Sub Worksheet_Change(ByVal Target As Range)
8     On Error Resume Next
9     If Not Intersect(Target, Range("G108:G111" )) Is Nothing Then 'On Change Of Custom Field
10        Time
11        Dim FieldTypeNumber As Long
12        FieldTypeNumber = Application.WorksheetFunction.Match(Target.Value, Sheet4.Range(
13            "FieldTypes" ), 0)
14        If FieldTypeNumber <> 0 Then
15            If FieldTypeNumber <> 5 Then
16                Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation.Delete 'Delete
17                Any possible Data validation
18                Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).NumberFormat = Sheet4.Range(
19                    "B" & 119 + FieldTypeNumber).Value
20            Else:
21                With Sheet1.Range(Sheet4.Range("B" & Target.Row).Value).Validation
22                    .Delete
23                    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween,
24                        Formula1:="=CustomList" & Target.Row - 107
25                    .IgnoreBlank = True
26                    .InCellDropdown = True
27                    .ShowInput = True
28                    .ShowError = True
29                End With
30            End If
31        End If
32    End If
33 End Sub
34
35 Private Sub Worksheet_Deactivate()
36     Range("B3" ).Value = 4
37     Admin_SwitchHorizontalTabs
38 End Sub
39
40 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
41     If Target.Count > 1 Then Exit Sub
42     If Not Intersect(Target, Range("D4:K4" )) Is Nothing Then
43         Range("B3" ).Value = Target.Column
44         Range("F2" ).Select
45         Admin_SwitchHorizontalTabs
46     End If
47 End Sub
```

A

ActiveWindow, [20](#)
Add, [20](#)
Admin_SwitchHorizontalTabs, [20](#)
AlertStyle, [20](#)
Application, [20](#)

C

Column, [20](#)
Count, [20](#)

D

Delete, [20](#)

E

Explicit, [20](#)

F

FieldTypeNumber, [20](#)
Formula1, [20](#)

I

IgnoreBlank, [20](#)
InCellDropdown, [20](#)
Intersect, [20](#)

M

Match, [20](#)

N

NumberFormat, [20](#)

O

Operator, [20](#)

R

Range, [20](#)
Row, [20](#)

S

ScrollRow, [20](#)
Sheet1, [20](#)
Sheet4, [20](#)
ShowError, [20](#)
ShowInput, [20](#)

T

Target, [20](#)

V

Validation, [20](#)
Value, [20](#)

W

Worksheet_Activate, [20](#)
Worksheet_Change, [20](#)
Worksheet_Deactivate, [20](#)
Worksheet_SelectionChange, [20](#)
WorksheetFunction, [20](#)

X

xlBetween, [20](#)
xlValidAlertStop, [20](#)
xlValidateList, [20](#)

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
```

```
        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
```

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

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

```
                Sheet7.Cells(Range("B3" ).Value, Cells(Target.Row, Target.Column + 23).Value).
                Value = "Yes"
```

```
                Sheet7.Range("AA" & Range("B3" ).Value).Value = Range("J10" ).Value + Range(
                    "L10" ).Value - Range("B11" ).Value 'Set Reminder On Date & Time
```

```
            Else: Sheet7.Cells(Range("B3" ).Value, Cells(Target.Row, Target.Column + 23).Value).
            Value = "No"
```

```
            End If
```

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

```
            Sheet7.Cells(Range("B3" ).Value, Cells(Target.Row, Target.Column + 23).Value).Value
            = Target.Value
```

```
        End If
```

```
        '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
```

```
End Sub
```

```
Private Sub Worksheet_Deactivate()
```

```
    Dim TP As Shape
```

```
    On Error Resume Next
```

```
    Set TP = Shapes("TimePickGrp" )
```

```
    On Error GoTo 0
```

```
    If TP Is Nothing Then 'Shape Deleted
```

```

1 2
48 Application.EnableEvents = False
49 On Error Resume Next
50 Application.Undo
51 Application.EnableEvents = True
52 End If
53 End Sub
54
55 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
56 If Target.Count > 1 Then Exit Sub
57 'For Pop-up Calendar
58 Dim Cal As Shape 'Calendar
59 On Error Resume Next
60 Set Cal = Shapes("Calendar" )
61 On Error GoTo 0
62 If Cal Is Nothing Then 'Shape Deleted
63 Application.EnableEvents = False
64 On Error Resume Next
65 Application.Undo
66 On Error GoTo 0
67 Application.EnableEvents = True
68 End If
69 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
70 CheckForSheet
71 CalendarShow
72 Else:
73 CheckForSheet
74 CalendarHide
75 End If
76
77 'For Pop-up Date Picker
78 Dim TP As Shape 'Time Picker
79 On Error Resume Next
80 Set TP = Shapes("TimePickGrp" )
81 On Error GoTo 0
82 If TP Is Nothing Then 'Shape Deleted
83 Application.EnableEvents = False
84 On Error Resume Next
85 Application.Undo
86 On Error GoTo 0
87 Application.EnableEvents = True
88 End If
89 If Not Intersect(Target, Range("L10,L11" )) Is Nothing Then
90 'If Sheet is Protected, add Unprotect Code here such as ActiveSheet.Unprotect
>Password"
91 TPShow
92 Else:
93 Shapes("TimePickGrp" ).Visible = msoFalse
94 End If
95
96 'On Table Selection
97 If Not Intersect(Target, Range("D27:N9999" )) Is Nothing And Range("D" & Target.Row).
Value <> Empty Then
98 Range("B10" ).Value = Target.Row 'Add in Selected Row
99 Range("B4" ).Value = Range("O" & Target.Row).Value 'Add In Event ID
100 Event_Load
101 End If
102
103 'On Recurring Selection
104 If Not Intersect(Target, Range("D10" )) Is Nothing Then
105 With ActiveCell
1 2 3

```

```
1 2 3
106 If .Value = "p Recurring" Then .Value = "o Recurring" Else: .Value = "p
      Recurring"
107 .Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
108 .Characters(Start:=2, Length:=16).Font.Name = "Calibri"
109 End With
110 Range("F11").Select
111 End If
112
113 'On Reminders selection
114 If Not Intersect(Target, Range("D13")) Is Nothing Then
115 With ActiveCell
116 If .Value = "p Reminders" Then .Value = "o Reminders" Else: .Value = "p
      Reminders"
117 .Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
118 .Characters(Start:=2, Length:=15).Font.Name = "Calibri"
119 If .Value = "p Reminders" Then
120 Sheet5.Shapes("EventRemindIcons").Visible = msoCTrue
121 Else:
122 Sheet5.Shapes("EventRemindIcons").Visible = msoFalse
123 Range("L13").Value = "No"
124 End If
125 End With
126 Range("I13").Select
127 End If
128 End Sub
129
130
```

A

ActiveCell, [23, 24](#)
Address, [22](#)
Application, [23](#)

C

Cal, [23](#)
CalendarHide, [23](#)
CalendarShow, [23](#)
Cancel, [22](#)
CellAdd, [22](#)
Cells, [22](#)
Characters, [24](#)
CheckForSheet, [23](#)
Column, [22](#)
Count, [23](#)

E

Empty, [22, 23](#)
EnableEvents, [23](#)
EntireRow, [22](#)
Event_Load, [23](#)
Explicit, [22](#)

F

Find, [22](#)
Font, [24](#)
FoundMapRng, [22](#)

H

Hidden, [22](#)

I

InStr, [22](#)
Intersect, [22-24](#)
IsNumeric, [22](#)

L

Length, [24](#)

M

MapRng, [22](#)
msoCTrue, [24](#)
msoFalse, [23, 24](#)

N

Name, [24](#)

R

Range, [22-24](#)
Row, [22, 23](#)

S

SelRow, [22](#)
Shape, [22, 23](#)
Shapes, [22-24](#)
Sheet2, [22](#)
Sheet5, [22, 24](#)
Sheet7, [22](#)
Start, [24](#)

T

Target, [22-24](#)
TP, [22, 23](#)
TPShow, [23](#)

U

Undo, [23](#)

V

Value, [22-24](#)
Visible, [23, 24](#)

W

Worksheet_BeforeDoubleClick, [22](#)
Worksheet_Change, [22](#)
Worksheet_Deactivate, [22](#)
Worksheet_SelectionChange, [23](#)

X

xlValues, [22](#)
xlWhole, [22](#)

1 Option Explicit

2

E

Explicit, [26](#)

1 Option Explicit

2

E

Explicit, [28](#)


```
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, [30](#)
ActiveWindow, [30](#)
AddCol, [30](#)
Address, [30](#)
Application, [30](#)
AutoFilter, [30](#)
AutoFilterMode, [30](#)

C

Cells, [30](#)
ClearComments, [30](#)
Column, [30](#)
Count, [30](#)
Criteria1, [30](#)

D

DisplayCommentIndicator, [30](#)

E

Empty, [30](#)
Explicit, [30](#)

F

Field, [30](#)

I

Intersect, [30](#)

R

Range, [30](#)
Row, [30](#)

S

Sched_AddComment_Event, [30](#)
Sched_AddComment_Holiday, [30](#)
Sched_MonthView, [30](#)
Sched_Refresh, [30](#)
Sched_ScrollToRow, [30](#)
Sched_WeekView, [30](#)
ScrollRow, [30](#)
ShowAllData, [30](#)

T

Target, [30](#)

V

Value, [30](#)

W

Worksheet_Activate, [30](#)
Worksheet_Change, [30](#)
Worksheet_SelectionChange, [30](#)

X

xlCommentIndicatorOnly, [30](#)

1 Option Explicit

2

E

Explicit, [32](#)

```
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 End Sub
10
11 Private Sub Workbook_Deactivate()
12     SetOnkey (False)
13 End Sub
14
15 Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel
16     As Boolean)
17     On Error Resume Next
18     Application.CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
19     Application.CommandBars("Cell" ).Controls("Go To Event" ).Delete
20     On Error GoTo 0
21     If ActiveSheet.CodeName = "Sheet8" Then RightClick_CallEvents
22 End Sub
```

A

ActiveSheet, [34](#)
Application, [34](#)

C

Cancel, [34](#)
CodeName, [34](#)
CommandBars, [34](#)
Controls, [34](#)

D

Delete, [34](#)

E

Explicit, [34](#)

R

Range, [34](#)
RightClick_CallEvents, [34](#)

S

SetOnkey, [34](#)
Sh, [34](#)

T

Target, [34](#)

W

Workbook_Activate, [34](#)
Workbook_Deactivate, [34](#)
Workbook_SheetBeforeRightClick, [34](#)

```
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, [36](#)
ClsBtn_Click, [36](#)
CreateBtn_Click, [36](#)
CreatePayPeriod, [36](#)
CreatePayPeriods, [36](#)

E

Explicit, [36](#)

H

Hide, [36](#)


```
1  
2 Option Explicit  
3  
4 Private Sub ToggleButton1_Click()  
5     CheckUser  
6 End Sub
```

C

CheckUser, [38](#)

E

Explicit, [38](#)

T

ToggleButton1_Click, [38](#)

```

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
65     xlPasteFormulas
66     .Range("A" & FormulaRow + 1).Copy
67     .Range("E" & StartRow + 1 & ":E" & StartRow + PayrollQty - 1).PasteSpecial
68     xlPasteFormulas
69     .Range("B14" ).Copy
70     .Range("D" & StartRow & ":D" & StartRow + PayrollQty - 1).PasteSpecial
71     xlPasteFormulas 'Create Pay Period Names
72 End With
73 Application.CutCopyMode = False
74 End Sub
75
76 Sub ShowPayPeriodForm()
77     CreatePayPeriod.Show
78 End Sub
```

A

AllowMultiSelect, [40](#)
Application, [40](#), [41](#)

B

BrowseEmployeeFolder, [40](#)
BrowseRemoteTimeClockFolder, [40](#)

C

ClearContents, [40](#)
Copy, [41](#)
CreatePayPeriod, [40](#), [41](#)
CreatePayPeriods, [40](#)
CutCopyMode, [41](#)

E

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

F

FileDialog, [40](#)
FormulaRow, [40](#), [41](#)
Frequency, [40](#)

H

Hide, [40](#)

M

MsgBox, [40](#)
msoFileDialogFolderPicker, [40](#)

N

NextDate, [40](#)
NoSelection, [40](#)

P

PasteSpecial, [41](#)
PayrollQty, [40](#), [41](#)

R

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

S

SelectedItems, [40](#)
Sheet4, [40](#)
Show, [40](#), [41](#)
ShowPayPeriodForm, [41](#)
StartDate, [40](#), [41](#)
StartRow, [40](#), [41](#)
strPath, [40](#)

T

Title, [40](#)

V

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

X

xlPasteFormulas, [41](#)
xlUp, [40](#)

```
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        .Range("5:504").EntireRow.Hidden = True
14        FirstRow = 5 + ((SelCol - 4) * 100)
15        .Range(FirstRow & ":" & FirstRow + 99).EntireRow.Hidden = False
16    End With
17    ResetCalc
18 End Sub
```

A

Admin_SwitchHorizontalTabs, 43

E

EntireRow, 43

Explicit, 43

F

FirstRow, 43

H

Hidden, 43

M

msoCTrue, 43

msoFalse, 43

R

Range, 43

ResetCalc, 43

S

SelCol, 43

Shapes, 43

Sheet4, 43

StopCalc, 43

V

Value, 43

Visible, 43

```

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

—, 47

A

ActiveCell, 45-47
ActiveSheet, 45
Address, 46
Application, 45-47
Array, 45

B

bFound, 46, 47

C

Cells, 47
CodeName, 45
Column, 47
Count, 46

D

do_nothing, 45

E

Empty, 46
EnableEvents, 46, 47
ExitSub, 46

G

GetColLtr, 45-47
GetTabOrder, 45, 46

I

i, 46, 47
iAdjust, 46
iDirection, 46
If, 46
iPos, 45
IsError, 46
iSign, 46, 47
IsNumeric, 45

L

LBound, 46
lItems, 46
lRowClosest, 46, 47
lRowTest, 46, 47

M

m, 46
Match, 46
Mid, 45

O

OnKey, 45, 46

R

Range, 46, 47
Row, 45, 47
Rows, 46

S

sActiveCol, 46, 47
sAddr, 45
SetOnkey, 45
state, 45
sTest, 45

T

TabRange, 46

U

UBound, 46
UpOrDownArrow, 46

V

vTabOrder, 46, 47

X

xLDown, 46
xLNext, 46
xLPrevious, 46
xLUp, 46

```
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, [49](#)

C

Calculation, [49](#)

E

EnableEvents, [49](#)

Explicit, [49](#)

R

ResetCalc, [49](#)

S

ScreenUpdating, [49](#)

StopCalc, [49](#)

X

xlCalculationAutomatic, [49](#)

xlCalculationManual, [49](#)

```

1  Option Explicit
2
3
4  Sub Attach_Add()
5      Dim FileFldr As FileDialog
6      Dim FileName, OrigFilePath, FileType, DestFileFolder, DestFilePath, EmpID, EmpName As String
7      Dim LastAttRow As Long
8      DestFileFolder = Sheet4.Range("EmplAttachFolder").Value
9      If DestFileFolder = "" Then 'If user has not added a main folder yet
10         MsgBox "Before attaching any files for this Employee, please select a dedicated" &
11             vbCrLf & "folder to store the individual Employee Attachments"
12         Exit Sub
13     End If
14     EmpID = Sheet1.Range("J2").Value 'Employee ID
15     EmpName = Replace(Replace(Sheet1.Range("F2").Value, ",", "_"), " ", "") 'Employee Name
16     Set FileFldr = Application.FileDialog(msoFileDialogFilePicker)
17     With FileFldr
18         .Title = "Select file to attach"
19         .Filters.Add "All Files", "*.*"
20         If .Show <> -1 Then GoTo NoSelection
21         OrigFilePath = .SelectedItems(1)
22         FileName = Dir(OrigFilePath)
23
24         FileType = Right(FileName, 5 - InStr(1, Right(FileName, 5), "."))
25     End With
26
27     'Check if Employee Folder exists, if not, add it.
28     If Len(Dir(DestFileFolder & "\" & EmpName, vbDirectory)) = 0 Then
29         On Error GoTo BadFolder
30         Mkdir DestFileFolder & "\" & EmpName
31     End If
32     DestFilePath = DestFileFolder & "\" & EmpName & "\" & FileName
33     FileCopy OrigFilePath, DestFilePath 'Copy user selected file into appropriate Employee folder
34     'Kill (OrigFilePath) 'If you want to delete the original file in the original location
35
36     With Sheet3
37         LastAttRow = .Range("D9999").End(xlUp).Row + 1 'First Avail. Attachment Row
38         .Range("D" & LastAttRow).Value = EmpID 'EmployeeID
39         .Range("E" & LastAttRow).Value = Sheet1.Range("F2").Value 'Employee Name
40         .Range("F" & LastAttRow).Value = FileName
41         .Range("G" & LastAttRow).Value = FileType
42         .Range("H" & LastAttRow).Value = DestFilePath
43         .Range("I" & LastAttRow).Value = Application.UserName
44         .Range("J" & LastAttRow).Value = Now
45         .Range("K" & LastAttRow).Value = "=Row()"
46     End With
47
48     Attach_Refresh 'Refresh List To Display New Attachment
49     Sheet1.Range("E19").Select 'Select First Item to Display Thumbnail (if any)
50     Attach_DisplayThumbnail
51     NoSelection:
52     Exit Sub
53     BadFolder:
54     MsgBox "Please select a correct folder to add attachments"
55     Sheet4.Activate
56     BrowseEmployeeFolder
57 End Sub
58 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:D99999").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"  
      & FiltRow).Value 'Copy Over Attachment Info  
173      AttRow = AttRow + 1  
174      If AttRow = 25 Then Exit Sub 'Exit If Table is Full  
175      Next FiltRow  
176      End With  
177      NoData:  
178  
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, [51](#), [53](#)
ActiveCell, [52](#)
Add, [51](#), [52](#)
AdvancedFilter, [53](#)
Application, [51](#)
Attach_Add, [51](#)
Attach_Delete, [51](#)
Attach_DisplayThumbnail, [51](#), [52](#)
Attach_Filter, [53](#)
Attach_FilterClear, [53](#)
Attach_Open, [53](#)
Attach_Refresh, [51-53](#)
Attach_ShowAll, [54](#)
AttRow, [52-54](#)
AttTableRow, [52](#)

B

BadFolder, [51](#)
BrowseEmployeeFolder, [51](#)

C

ClearContents, [53](#)
CopyToRange, [53](#)
CriteriaRange, [53](#)

D

Delete, [52](#)
DestFileFolder, [51](#)
DestFilePath, [51](#)
Dir, [51](#)
DisplayAsIcon, [52](#)

E

EmpID, [51-53](#)
EmpName, [51](#)
Empty, [52](#), [53](#)
EntireRow, [52](#), [53](#)
Explicit, [51](#)

F

FileCopy, [51](#)
FileDialog, [51](#)
FileFldr, [51](#)
FileName, [51](#), [52](#)
FilePath, [52](#)
FileType, [51](#), [52](#)
Filters, [51](#)
FiltRow, [53](#), [54](#)
FollowHyperlink, [53](#)

H

Height, [52](#)
Hidden, [53](#)

I

IncrementLeft, [52](#)
IncrementTop, [52](#)
Insert, [52](#)
InStr, [51](#)

L

LastAttRow, [51](#), [53](#)
LastFiltRow, [53](#)
LCase, [52](#)
Left, [52](#)
Len, [51](#)
Link, [52](#)
LockAspectRatio, [52](#)

M

MkDir, [51](#)
MsgBox, [51-53](#)
msoCTrue, [52](#)
msoFileDialogFilePicker, [51](#)

N

Name, [52](#)
NoAttach, [52](#), [53](#)
NoData, [53](#), [54](#)
NoSelection, [51](#)
Now, [51](#)

O

OLEObjects, [52](#)
OrigFilePath, [51](#)

P

Pictures, [52](#)

R

Range, [51-54](#)
Replace, [51](#)
Right, [51](#)
Row, [51-53](#)

S

SelectedItems, [51](#)
Shapes, [52](#)
Sheet1, [51-54](#)
Sheet3, [51-54](#)
Sheet4, [51](#)
Show, [51](#)
ShowAllData, [53](#)

T

ThisWorkbook, [53](#)
ThumbType, [52](#)
Title, [51](#)
Top, [52](#)

U

Unique, [53](#)
UserName, [51](#)

V

Value, [51-54](#)
vbCrLf, [51](#)
vbDirectory, [51](#)
Visible, [52](#)

W

Width, [52](#)

X

xlFilterCopy, [53](#)
xlFilterInPlace, [53](#)
xlUp, [51](#), [53](#)

```

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, [56](#), [57](#)
ActiveCell, [56](#)
ActiveSheet, [57](#)
ActiveWindow, [56](#), [57](#)
ActRow, [56](#)
Address, [56](#)
AddTextbox, [56](#)
Alignment, [56](#)
AutoShapeType, [57](#)

C

Cells, [56](#), [57](#)
Collate, [57](#)
Column, [56](#)
Copies, [57](#)

D

Delete, [57](#)

E

EmpIDCard_CreateTextBox, [56](#)
EmpIDCard_Customize, [56](#)
EmpIDCard_FieldChange, [56](#)
EmpIDCard_Hide, [56](#)
EmpIDCard_HideCustom, [56](#)
EmpIDCard_HidePic, [56](#), [57](#)
EmpIDCard_Print, [57](#)
EmpIDCard_Show, [57](#)
EmpIDCard_ShowPic, [56](#), [57](#)
EntireColumn, [56](#), [57](#)
Explicit, [56](#)

F

Fill, [56](#)
Formula, [56](#)

H

Height, [57](#)
Hidden, [56](#), [57](#)

I

IgnorePrintAreas, [57](#)
IncrementLeft, [57](#)
IncrementTop, [57](#)
Insert, [57](#)

L

Left, [57](#)
Line, [56](#)
LockAspectRatio, [57](#)

M

msoAlignCenter, [56](#)
msoCTrue, [56](#), [57](#)
msoFalse, [56](#), [57](#)
msoShapeRoundedRectangle, [57](#)
msoTextOrientationHorizontal, [56](#)
msoTrue, [57](#)

N

Name, [56](#), [57](#)
NoPic, [57](#)

P

PageSetup, [57](#)
ParagraphFormat, [56](#)
PicPath, [57](#)
Pictures, [57](#)

PrintArea, [57](#)
PrintOut, [57](#)

R

Range, [56](#), [57](#)
Row, [56](#)

S

ScrollColumn, [56](#), [57](#)
Selection, [56](#)
Shape, [56](#)
ShapeRange, [57](#)
Shapes, [56](#), [57](#)
Sheet1, [56](#), [57](#)

T

text, [56](#)
TextBox, [56](#)
TextBoxName, [56](#)
TextFrame2, [56](#)
TextRange, [56](#)
Top, [57](#)

V

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

```

1 Option Explicit
2
3 Sub Emp_SaveNew()
4     Dim EmpRow As Long
5     Dim EmpCol As Long
6     With Sheet1
7         If .Range("F6").Value = Empty Then
8             MsgBox "Please enter a Last Name for the Employee"
9             .Range("F6").Select
10            Exit Sub
11        End If
12        EmpRow = Sheet2.Range("A99999").End(xlUp).Row + 1 'First Avail Row
13        Sheet2.Range("A" & EmpRow).Value = Application.WorksheetFunction.Max(Sheet2.Range(
14            "EmployeeID" )) + 1
15        For EmpCol = 2 To 33
16            Sheet2.Cells(EmpRow, EmpCol).Value = .Range(Sheet2.Cells(1, EmpCol).Value).Value
17        Next EmpCol
18        .Range("F2").Value = .Range("F6").Value & ", " & .Range("I6").Value 'Add in Last
19        Name, First Name
20        .Shapes("NewEmpGrp").Visible = msoFalse
21        .Shapes("ExistEmpGrp").Visible = msoCTrue
22        .Range("B1").Value = False 'Set Empl Load To False
23        .Range("B6").Value = False 'Set New Employee To False
24    End With
25    Empl_SortByName 'Sort Employee List
26    Sheet8.Range("D8:D314").Formula = Sheet8.Range("D8").Formula 'Update Employee Schedule
27    Names
28 End Sub
29
30 Sub Empl_Archive()
31     Dim EmplRow As Long
32     Dim ArchiveRow As Long
33     Dim EmplName As String
34     If MsgBox("Are you sure you want to Archive this employee?" & vbCrLf & "Their name will
35         no longer be available in the drop down list", vbYesNo, "Archive Employee") = vbNo Then
36         Exit Sub
37     With Sheet1
38         If .Range("B4").Value = Empty Then
39             MsgBox "Please select an employee to Archive"
40             Exit Sub
41         End If
42         EmplName = .Range("F2").Value 'Employee Name
43         EmplRow = .Range("B4").Value 'Employee Row
44         ArchiveRow = Sheet6.Range("A99999").End(xlUp).Row + 1 'First Available Archive Row
45         .Range("H14").Value = "Inactive" 'You can change this to whatever you want
46         Sheet6.Range("A" & ArchiveRow & ":AG" & ArchiveRow).Value = Sheet2.Range("A" &
47             EmplRow & ":AG" & EmplRow).Value 'Copy Over Employee Data
48         Sheet2.Range(EmplRow & ":" & EmplRow).EntireRow.Delete
49         Sheet1.Range("F2").Value = Sheet2.Range("BC4").Value 'Reset to display First
50         Employee
51         'Update Employee List in the Schedule
52         Sheet8.Range("D8").Formula = "'Employee List'!BC4"
53         Sheet8.Range("D8:D314").Formula = Sheet8.Range("D8").Formula 'Update Employee
54         Schedule Names
55         MsgBox EmplName & " has been archived"
56     End With
57 End Sub
58
59 Sub Empl_CancelNew()
60
61

```

```

1
54 Sheet1.Range("F2" ).Value = Sheet2.Range("BC4" ).Value 'Reset to display First Employee
55 End Sub
56
57 Sub Empl_EventsRefresh()
58 Dim LastEventRow As Long
59 Dim LastFiltRow As Long
60 With Sheet1
61 'Load Events List Into Events Sheet Using Advanced Filter & Sort
62 Sheet7.Range("BA2:BK2,BA5:BK9999" ).ClearContents 'Clear any Previous Results
63 Sheet7.Range("BA2" ).Value = .Range("J2" ).Value 'Employee ID
64 LastEventRow = Sheet7.Range("A9999" ).End(xlUp).Row
65 On Error GoTo NoEvents
66 Sheet7.Range("A4:W" & LastEventRow).AdvancedFilter xlFilterCopy, CriteriaRange:=Sheet7
        .Range("BA1:BA2" ), CopyToRange:=Sheet7.Range("BA4:BK4" ), Unique:=True
67 LastFiltRow = Sheet7.Range("BA9999" ).End(xlUp).Row
68 If LastFiltRow < 5 Then GoTo NoEvents
69 'Sort events from Newest To Oldest
70 Sheet7.Sort.SortFields.Clear
71 Sheet7.Sort.SortFields.Add Key:=.Range("BD5" ), SortOn:=xlSortOnValues, Order:=
        xlDescending, DataOption:=xlSortNormal
72 With Sheet7.Sort
73 .SetRange Range("BA5:BK" & LastFiltRow)
74 .Apply
75 End With
76 If LastFiltRow > 21 Then LastFiltRow = 21 'Set Maximum of 17 Rows
77 .Range("E188:H" & LastFiltRow + 183).Value = Sheet7.Range("BB5:BE" & LastFiltRow).
        Value 'Copy First Section
78 .Range("AE188:AF188" ).Copy 'Copy Formulas To Combine Frequency & Start On Fields
79 .Range("I188:J" & LastFiltRow + 183).PasteSpecial xlPasteFormulas 'Paste Down
        Formulas
80 .Range("K188:L" & LastFiltRow + 183).Value = Sheet7.Range("BJ5:BK" & LastFiltRow).
        Value 'Copy Last Section
81 Application.CutCopyMode = xlCopy
82 NoEvents:
83 End With
84 End Sub
85
86
87 Sub Empl_LeaveRefresh()
88 Dim LastLeaveRow As Long
89 Dim LastFiltRow As Long
90 With Sheet1
91 Sheet11.Range("AA3,AA5:ADI9999" ).ClearContents 'Clear any Previous Results
92 Sheet11.Range("AA3" ).Value = .Range("J2" ).Value 'Employee ID
93 LastLeaveRow = Sheet11.Range("A9999" ).End(xlUp).Row
94 Sheet11.Range("A4:H" & LastLeaveRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
        Sheet11.Range("AA2:AA3" ), CopyToRange:=Sheet11.Range("AA4:AE4" ), Unique:=True
95 LastFiltRow = Sheet11.Range("AA9999" ).End(xlUp).Row
96 If LastFiltRow < 5 Then GoTo NoLeave
97
98 If LastFiltRow > 16 Then LastFiltRow = 16 'Set Maximum of 12 Rows
99 .Range("E168:H" & LastFiltRow + 163).Value = Sheet11.Range("AA5:AD" & LastFiltRow).
        Value 'Copy Date, Start & End Times ' Add in Leave Type, Paid, Carryover, Allowance
100 .Range("L168:L" & LastFiltRow + 163).Value = Sheet11.Range("AE5:AE" & LastFiltRow).
        Value 'Copy Date, Start & End Times 'Add In Row #'s
101 NoLeave:
102 End With
103 End Sub
104
105 Sub Empl_Load()
106 Dim EmpRow As Long
1

```

```

1
107 Dim EmpCol As Long
108 With Sheet1
109     If .Range("B5").Value = Empty Then
110         MsgBox "Please enter a valid Employee from the drop down list"
111         Exit Sub
112     End If
113     StopCalc
114     .Range("B1").Value = True 'Set Employee Load To True
115     EmpRow = .Range("B5").Value
116     .Range(
117         "J2:F6:G6,I6,F8:I8,F10:G10,I10,K10,F12,H12:I12,F14,H14:I14,K14,F16,H16:I16,K16,G66,J66,G
118         68,J68,G70,J70,G73,J73,G75,J75,G86,J86,G88,J88,G182,I182,K182,E183").ClearContents
119     .Range(
120         "F26:G26,I26,K26,E19:K24,E28:L44,F46:G46,I46,K46,E48:L64,G93:K104,G108:K119,G128:K139,F1
121         48:J154,L148:L154,E168:H179,L168:HL179,E188:K204").ClearContents
122     For EmpCol = 1 To 33
123         .Range(Sheet2.Cells(1, EmpCol).Value).Value = Sheet2.Cells(EmpRow, EmpCol).Value
124     Next EmpCol
125     .Shapes("NewEmpGrp").Visible = msoFalse
126     .Shapes("ExistEmpGrp").Visible = msoCTrue
127     On Error Resume Next
128     .Shapes("AttachThumb").Delete 'Remove Attachment thumbnail, If any
129     On Error GoTo 0
130     .Range("B6").Value = False 'Set New Employee To False
131     .Range("E4").Select 'Load General Info Tab
132     If .Range("B2").Value = 5 Then Show_EmplPic 'Only when First Tab is selected
133     Attach_Refresh
134     Empl_EventsRefresh 'Refresh Employee Events
135     Empl_TimeclockRefresh 'Refresh Time Clock Entries
136     Empl_LeaveRefresh 'Refresh Employee Leave Items
137     .Range("B1").Value = False 'Set Employee Load To False
138 End With
139 ResetCalc
140 End Sub
141
142 Sub Empl_New()
143     With Sheet1
144         .Range("B1").Value = True 'Set Employee Load To False
145         .Shapes("NewEmpGrp").Visible = msoCTrue
146         .Shapes("ExistEmpGrp").Visible = msoFalse
147         .Shapes("DefaultPicture").Visible = msoCTrue
148         On Error Resume Next
149         .Shapes("EmplPic").Delete
150         On Error GoTo 0
151         .Range(
152             "F2:G2,J2,F6:G6,I6,F8:I8,F10:G10,I10,K10,F12,H12:I12,F14,H14:I14,K14,F16,H16:I16,K16,G66
153             ,J66,G68,J68,G70,J70,G73,J73,G75,J75,G86,J86,G88,J88,G182,I182,K182,E183").
154         ClearContents
155         .Range(
156             "F26:G26,I26,K26,E19:K24,E28:L44,F46:G46,I46,K46,E48:L64,G93:K104,G108:K119,G128:K139,F1
157             48:J154,L148:L154,E168:H179,L168:HL179,E188:K204").ClearContents
158         'Add New Employee Default Values
159         .Range("B6").Value = True 'Set New Employee To True
160         .Range("E4").Select 'Selects General Info Tab
161         .Range("F6").Select 'Select Last Name Field
162         .Range("B1").Value = False 'Set Employee Load To False
163     End With
164
1

```



```

1
159 End Sub
160
161 Sub Empl_SortByName()
162     'Sort Employee List
163     StopCalc
164     Dim EmpRow As Long
165     With Sheet2
166         EmpRow = .Range("A99999").End(xlUp).Row 'Last Row of value
167         .Sort.SortFields.Clear
168         .Sort.SortFields.Add Key:=.Range("B4"), SortOn:=xlSortOnValues, Order:=xlAscending,
169         DataOption:=xlSortNormal
170     With .Sort
171         .SetRange Range("A4:AG" & EmpRow)
172         .Apply
173     End With
174     End With
175     ResetCalc
176 End Sub
177
178
179 Sub Empl_TimeclockRefresh()
180     Dim LastTimeclockRow As Long
181     Dim LastFiltRow As Long
182     With Sheet1
183         Sheet9.Range("AA3:AI3,AA5:AAI99999").ClearContents 'Clear any Previous Results
184         Sheet9.Range("AB3").Value = .Range("J2").Value 'Employee ID
185         LastTimeclockRow = Sheet9.Range("A99999").End(xlUp).Row
186         Sheet9.Range("A4:I" & LastTimeclockRow).AdvancedFilter xlFilterCopy, CriteriaRange:=
187         Sheet9.Range("AA2:AA3"), CopyToRange:=Sheet9.Range("AA4:AF4"), Unique:=True
188         LastFiltRow = Sheet9.Range("AA99999").End(xlUp).Row
189         If LastFiltRow < 5 Then GoTo NoTimeclock
190         'Sort TimeClock Entries Newest To Oldest
191         Sheet9.Sort.SortFields.Clear
192         Sheet9.Sort.SortFields.Add Key:=.Range("AB5"), SortOn:=xlSortOnValues, Order:=
193         xlDescending, DataOption:=xlSortNormal
194     With Sheet9.Sort
195         .SetRange Range("AA5:AI" & LastFiltRow)
196         .Apply
197     End With
198     If LastFiltRow > 21 Then LastFiltRow = 21 'Set Maximum of 17 Rows
199     StopCalc
200     Sheet9.Range("AG1:AI1").Copy
201     Sheet9.Range("AG5:AI" & LastFiltRow).PasteSpecial xlPasteFormulas
202     ResetCalc
203     .Range("E28:G" & LastFiltRow + 23).Value = Sheet9.Range("AA5:AC" & LastFiltRow).Value
204     'Copy Date, Start & End Times
205     .Range("H28:J" & LastFiltRow + 23).Value = Sheet9.Range("AG5:AI" & LastFiltRow).Value
206     'Copy Total Break, Reg. Hours & OT Hours
207     .Range("K28:K" & LastFiltRow + 23).Value = Sheet9.Range("AF5:AG" & LastFiltRow).Value
208     'Copy Notes
209     Application.CutCopyMode = xlCopy
210     NoTimeclock:
211 End With
212 End Sub

```

A

Add, [60](#), [62](#)
AdvancedFilter, [60](#), [62](#)
Application, [59](#), [60](#), [62](#)
Apply, [60](#), [62](#)
ArchiveRow, [59](#)
Attach_Refresh, [61](#)

C

Cells, [59](#), [61](#)
Clear, [60](#), [62](#)
ClearContents, [60-62](#)
Copy, [60](#), [62](#)
CopyToRange, [60](#), [62](#)
CriteriaRange, [60](#), [62](#)
CutCopyMode, [60](#), [62](#)

D

DataOption, [60](#), [62](#)
Delete, [59](#), [61](#)

E

Emp_SaveNew, [59](#)
EmpCol, [59](#), [61](#)
Empl_Archive, [59](#)
Empl_CancelNew, [59](#)
Empl_EventsRefresh, [60](#), [61](#)
Empl_LeaveRefresh, [60](#), [61](#)
Empl_Load, [60](#)
Empl_New, [61](#)
Empl_SortByName, [59](#), [62](#)
Empl_TimeclockRefresh, [61](#), [62](#)
EmplName, [59](#)
EmplRow, [59](#)
EmpRow, [59-62](#)
Empty, [59](#), [61](#)
EntireRow, [59](#)
Explicit, [59](#)

F

Formula, [59](#)

K

Key, [60](#), [62](#)

L

LastEventRow, [60](#)
LastFiltRow, [60](#), [62](#)
LastLeaveRow, [60](#)
LastTimeclockRow, [62](#)

M

Max, [59](#)
MsgBox, [59](#), [61](#)
msoCTrue, [59](#), [61](#)
msoFalse, [59](#), [61](#)

N

NoEvents, [60](#)
NoLeave, [60](#)
NoTimeclock, [62](#)

O

Order, [60](#), [62](#)

P

PasteSpecial, [60](#), [62](#)

R

Range, [59-62](#)
ResetCalc, [61](#), [62](#)
Row, [59](#), [60](#), [62](#)

S

SetRange, [60](#), [62](#)
Shapes, [59](#), [61](#)
Sheet1, [59-62](#)
Sheet11, [60](#)
Sheet2, [59-62](#)
Sheet6, [59](#)
Sheet7, [60](#)
Sheet8, [59](#)
Sheet9, [62](#)
Show_EmplPic, [61](#)
Sort, [60](#), [62](#)
SortFields, [60](#), [62](#)
SortOn, [60](#), [62](#)
StopCalc, [61](#), [62](#)

U

Unique, [60](#), [62](#)

V

Value, [59-62](#)
vbCrLf, [59](#)
vbNo, [59](#)
vbYesNo, [59](#)
Visible, [59](#), [61](#)

W

WorksheetFunction, [59](#)

X

xlAscending, [62](#)
xlCopy, [60](#), [62](#)
xlDescending, [60](#), [62](#)
xlFilterCopy, [60](#), [62](#)
xlPasteFormulas, [60](#), [62](#)
xlSortNormal, [60](#), [62](#)
xlSortOnValues, [60](#), [62](#)
xlUp, [59](#), [60](#), [62](#)

```

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

D

DefaultPic, [64](#), [65](#)
Delete, [64](#)
DestFileFolder, [64](#)
DestFilePath, [64](#)
Dir, [64](#)

E

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

F

FileCopy, [64](#)
FileDialog, [64](#)
FileName, [64](#)
Filters, [64](#)

H

Height, [64](#)

I

IncrementLeft, [65](#)
IncrementTop, [65](#)
Insert, [64](#)

L

Left, [65](#)
Len, [64](#)
LockAspectRatio, [64](#)

M

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

N

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

O

OrigFilePath, [64](#)

P

PicFile, [64](#)
PicPath, [64](#)
Pictures, [64](#)

R

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

S

SelectedItems, [64](#)
ShapeRange, [64](#)
Shapes, [64](#), [65](#)
Sheet1, [64](#), [65](#)
Sheet4, [64](#)
Show, [64](#)
Show_EmplPic, [64](#)

T

Title, [64](#)
Top, [65](#)

V

Value, [64](#)
vbCrLf, [64](#)
vbDirectory, [64](#)
Visible, [64](#), [65](#)

```

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

```

1

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

A

ActiveCell, [68](#)
ActiveSheet, [67](#), [68](#)

C

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

E

Empl_SwitchHorizontalTabs, [67](#)
EmpTab_Events, [67](#)
EmpTab_GenInfo, [67](#)
EmpTab_Leave, [67](#)
EmpTab_PayDetails, [67](#)
EmpTab_PayHistory, [67](#)
EmpTab_Scheduling, [67](#)
EmpTab_TimeClock, [68](#)
Empty, [67](#)
EntireRow, [67](#), [68](#)
Explicit, [67](#)

F

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

H

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

M

msoCTrue, [67](#)
msoFalse, [67](#)

R

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

S

SelCol, [67](#)
SelRow, [68](#)
Shapes, [67](#)
Sheet1, [67](#), [68](#)
Show_EmplPic, [67](#)
StopCalc, [67](#)
SwitchVerticalTabs, [68](#)

V

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


```

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
13 Sub Event_AddEmailAttach()
14     Dim FileFldr As FileDialog
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         Sheet5.Range("F16" ).Value = .SelectedItems(1)
21     End With
22     NoSelection:
23 End Sub
24 Sub Event_AddNew()
25     With Sheet5
26         .Range("B1" ).Value = True 'Set Event Load To True
27         .Range("B2" ).Value = True 'Set New Event To True
28         .Range(
29             "B4,B6,E4,H4,K4,E6:N8,F10,G10,J10,L10,N10,J11,L11,F13,G13,N13,F15:I15,K15,M15,F16:M16,F1
30             7:N17,F18:N21" ).ClearContents
31         .Shapes("NewEventGrp" ).Visible = msoCTrue
32         .Shapes("ExistEventGrp" ).Visible = msoFalse
33         .Range("N4" ).Value = Now 'Set Current Time & Date
34         .Range("B1" ).Value = False 'Set Event Load To False
35         .Range("E4" ).Select
36     End With
37 End Sub
38 Sub Event_CancelNew()
39     Sheet5.Range("D27" ).Select
40 End Sub
41 Sub Event_Delete()
42     If MsgBox("Are you sure you want to delete this event?" , vbYesNo, "Delete Event" ) = vbNo
43     Then Exit Sub
44     With Sheet5
45         If .Range("B3" ).Value = Empty Then Exit Sub
46         EventRow = .Range("B3" ).Value 'Event Row
47         Sheet7.Range(EventRow & ":" & EventRow).EntireRow.Delete
48         Event_Refresh 'Refresh Events
49     End With
50 End Sub
51 Sub Event_Load()
52     Dim SelRow As Long
53     With Sheet5
54         SelRow = .Range("B10" ).Value
55         If .Range("B3" ).Value = Empty Then
56             MsgBox "Please select on an Event to display the Event details"
57             Exit Sub
58         End If

```

```

1 2
59 StopCalc
60 .Range("B1" ).Value = True 'Set Event Load To True
61 EventRow = .Range("B3" ).Value
62
63 For EventCol = 1 To 26
64 MapRng = Sheet7.Cells(1, EventCol).Value
65 If MapRng = "D10" Or MapRng = "D13" Then 'For Recurring & Reminders
66 If MapRng = "D10" Then
67 If Sheet7.Cells(EventRow, EventCol).Value = "Yes" Then .Range(MapRng).Value =
68 "p Recurring" Else: .Range(MapRng).Value = "o Recurring"
69 .Range(MapRng).Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
70 .Range(MapRng).Characters(Start:=2, Length:=16).Font.Name = "Calibri"
71 End If
72 If MapRng = "D13" Then
73 If Sheet7.Cells(EventRow, EventCol).Value = "Yes" Then .Range(MapRng).Value =
74 "p Reminders" Else: .Range(MapRng).Value = "o Reminders"
75 .Range(MapRng).Characters(Start:=1, Length:=1).Font.Name = "Wingdings"
76 .Range(MapRng).Characters(Start:=2, Length:=15).Font.Name = "Calibri"
77 End If
78 Else: 'For All other fields
79 .Range(MapRng).Value = Sheet7.Cells(EventRow, EventCol).Value 'Add Mapped Values
80 End If
81 Next EventCol
82 If .Range("D13" ).Value = "p Reminders" Then .Shapes("EventRemindIcons" ).Visible =
83 msoCTrue Else .Shapes("EventRemindIcons" ).Visible = msoFalse 'Show Hide Reminder
84 Icon Group
85 .Shapes("NewEventGrp" ).Visible = msoFalse
86 .Shapes("ExistEventGrp" ).Visible = msoTrue
87 .Range("B2" ).Value = False 'Set New Event To False
88 .Range("B1" ).Value = False 'Set Event Load To False
89 ResetCalc
90 End With
91 End Sub
92
93 Sub Event_Refresh()
94 Dim LastEventRow As Long
95 Dim LastFiltRow As Long
96 With Sheet5
97 StopCalc
98 'Load Events List Into Events Sheet Using Advanced Filter & Sort
99 .Range("D27:O99999" ).ClearContents 'Clear Existing Events list
100 Sheet7.Range("AD5:AQ99999" ).ClearContents 'Clear any Previous Results
101 LastEventRow = Sheet7.Range("A99999" ).End(xlUp).Row
102 DeleteFilters
103
104 Sheet7.Range("A4:AA" & LastEventRow).AdvancedFilter xlFilterCopy, , CopyToRange:=
105 Sheet7.Range("AD4:AQ4" ), Unique:=True
106 LastFiltRow = Sheet7.Range("AD99999" ).End(xlUp).Row
107 If LastFiltRow < 5 Then GoTo NoEvent
108 'Sort Events From Newest To Oldest
109 Sheet7.Sort.SortFields.Clear
110 Sheet7.Sort.SortFields.Add Key:=.Range("AF5" ), SortOn:=xlSortOnValues, Order:=
111 xlDescending, DataOption:=xlSortNormal
112 With Sheet7.Sort
113 .SetRange Range("AD5:AQ" & LastFiltRow)
114 .Apply
115 End With
116 .Range("D27:I" & LastFiltRow + 22).Value = Sheet7.Range("AD5:AI" & LastFiltRow).Value
117 'Copy First Section
118 .Range("A27:B27" ).Copy 'Copy Formulas To Combine Frequency & Start On Fields

```

1 2

```

1 2
112 .Range("J27:K" & LastFiltRow + 22).PasteSpecial xlPasteFormulas 'Paste Down Formulas
113 .Range("L27:O" & LastFiltRow + 22).Value = Sheet7.Range("AN5:AQ" & LastFiltRow).Value
    'Copy Second Section
114 Application.CutCopyMode = xlCopy
115 .Range("D27" ).Select
116 NoEvent:
117 ResetCalc
118 End With
119 End Sub
120 Sub Event_SaveNew()
121 With Sheet7
122 'Check for Required Fields
123 If Sheet5.Range("E4" ).Value = Empty Or Sheet5.Range("H4" ).Value = Empty Then
124     MsgBox "Event Name and Event Type are required fields for Events"
125     Exit Sub
126 End If
127 EventRow = .Range("A99999" ).End(xlUp).Row + 1 'First Available Event List Row
128 .Cells(EventRow, 1).Value = Sheet5.Range("B7" ).Value 'New Event ID
129 For EventCol = 2 To 23
130     If EventCol = 8 Or EventCol = 15 Then 'Reminders & Recurring Options
131         If InStr(Sheet5.Range(.Cells(1, EventCol).Value).Value, "p" ) > 0 Then .Cells(
            EventRow, EventCol).Value = "Yes" Else: .Cells(EventRow, EventCol).Value = "No"
132     Else:
133         .Cells(EventRow, EventCol).Value = Sheet5.Range(.Cells(1, EventCol).Value).Value
            'Event Field
134     End If
135 Next EventCol
136 End With
137 With Sheet5
138 .Shapes("NewEventGrp" ).Visible = msoFalse
139 .Shapes("ExistEventGrp" ).Visible = msoTrue
140 .Range("B2" ).Value = False 'Set New Event To False
141 .Range("B4" ).Value = Sheet7.Cells(EventRow, 1).Value 'New Event ID
142 Event_Refresh ' Reload Events
143 End With
144 End Sub

```

A

Add, [70](#), [71](#)
AdvancedFilter, [71](#)
Application, [70](#), [72](#)
Apply, [71](#)

C

Cells, [71](#), [72](#)
Characters, [71](#)
Clear, [71](#)
ClearContents, [70](#), [71](#)
Copy, [71](#)
CopyToRange, [71](#)
CutCopyMode, [72](#)

D

DataOption, [71](#)
Delete, [70](#)
DeleteFilters, [70](#), [71](#)

E

Empty, [70](#), [72](#)
EntireRow, [70](#)
Event_AddEmailAttach, [70](#)
Event_AddNew, [70](#)
Event_CancelNew, [70](#)
Event_Delete, [70](#)
Event_Load, [70](#)
Event_Refresh, [70-72](#)
Event_SaveNew, [72](#)
EventCol, [70-72](#)
EventRow, [70-72](#)
Explicit, [70](#)

F

FileDialog, [70](#)
FileFldr, [70](#)
Filters, [70](#)
Font, [71](#)

I

InStr, [72](#)

K

Key, [71](#)

L

LastEventRow, [71](#)
LastFiltRow, [71](#), [72](#)
Length, [71](#)

M

MapRng, [70](#), [71](#)
MsgBox, [70](#), [72](#)
msoCTrue, [70](#), [71](#)
msoFalse, [70-72](#)
msoFileDialogFilePicker, [70](#)
msoTrue, [71](#), [72](#)

N

Name, [71](#)
Names, [70](#)
NoEvent, [71](#), [72](#)
NoSelection, [70](#)
Now, [70](#)

O

Order, [71](#)

P

PasteSpecial, [72](#)

R

Range, [70-72](#)
ResetCalc, [71](#), [72](#)
Row, [71](#), [72](#)

S

SelectedItems, [70](#)
SelRow, [70](#)
SetRange, [71](#)
Shapes, [70-72](#)
Sheet5, [70-72](#)
Sheet7, [70-72](#)
Show, [70](#)
Sort, [71](#)
SortFields, [71](#)
SortOn, [71](#)
Start, [71](#)
StopCalc, [71](#)

T

Title, [70](#)

U

Unique, [71](#)

V

Value, [70-72](#)
vbNo, [70](#)
vbYesNo, [70](#)
Visible, [70-72](#)

X

xlCopy, [72](#)
xlDescending, [71](#)
xlFilterCopy, [71](#)
xlPasteFormulas, [72](#)
xlSortNormal, [71](#)
xlSortOnValues, [71](#)
xlUp, [71](#), [72](#)

```
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, [74](#)

D

delay, [74](#)

DoEvents, [74](#)

E

EmployIDNotFound, [74](#)

Explicit, [74](#)

F

Fill, [74](#)

I

i, [74](#)

S

Shapes, [74](#)

StartTime, [74](#)

T

Timer, [74](#)

Transparency, [74](#)

V

Visible, [74](#)

```
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, [76](#)
AllowMultiSelect, [76](#)
Application, [76](#)

B

BeregnerNm, [76](#)
BeregnerRng, [76](#)

E

EmailAdd, [76](#)
Explicit, [76](#)

F

FileDialog, [76](#)
Find, [76](#)
FoundBeregner, [76](#)

G

GetFolder, [76](#)

I

InitialFileName, [76](#)

M

MailToSelected, [76](#)
msoFileDialogFolderPicker, [76](#)

N

NextCode, [76](#)

P

PDFFldr, [76](#)

R

Range, [76](#)
Row, [76](#)

S

SelectedItems, [76](#)
SelRow, [76](#)
Sheet1, [76](#)
Sheet2, [76](#)
Show, [76](#)
sItem, [76](#)
strPath, [76](#)

T

Title, [76](#)

V

Value, [76](#)

X

xlValues, [76](#)
xlWhole, [76](#)


```
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
94     Sheet
95 End Sub
96 Sub CalendarShow()
97     With ActiveSheet
98         Set SelCell = Selection
99         'Check if active cell is a valid date
100        If IsDate(SelCell.Value) = True Then
101            Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
102        Else: 'If No Date or incorrect Date user current date
103            Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
104        End If
105        Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).Value)
106        'Set Month
107        Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).Value)
108        'Set Year
109        DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
110        UNGROUPCAL
111        If InStr(.Shapes("NextTri" ).OnAction, "!" ) <> 0 Or InStr(.Shapes("1Day" ).
112        DrawingObject.Formula, "]" ) <> 0 Then 'Run Workbook Link Remover and Cell Link
113        Replacement
114        MacroLinkRemover
115        CalFormulaReplacement
116    End If
117    GroupCal
118    On Error GoTo NoCal
119    With ActiveSheet.Shapes(DayName)
120        .Fill.ForeColor.RGB = RGB(252, 213, 180)
121        .TextFrame2.TextRange.Font.Bold = msoTrue

```

1 2 3

```

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

```

```

174
175 'Reassign Shape Links & Macros
176 With ActiveSheet
177     UnGroupCal
178     '.Unprotect
179     .Shapes("NextRec" ).OnAction = "" & ActiveWorkbook.Name & "!NextMonth"
180     .Shapes("NextTri" ).OnAction = "" & ActiveWorkbook.Name & "!NextMonth"
181     .Shapes("PrevRec" ).OnAction = "" & ActiveWorkbook.Name & "!PrevMonth"
182     .Shapes("PrevTri" ).OnAction = "" & ActiveWorkbook.Name & "!PrevMonth"
183     .Shapes("SetBtn" ).OnAction = "" & ActiveWorkbook.Name & "!ShowSettings"
184     .Shapes("MonthYear" ).DrawingObject.Formula = "=CalPopUp!A5"
185     DayCnt = 1
186     For RowCnt = 1 To 6
187         For ColCnt = 2 To 8
188             .Shapes(DayCnt & "Day" ).DrawingObject.Formula = "=CalPopUp!" & .Cells(RowCnt,
189             ColCnt).Address 'Assigned Linked Cell
190             .Shapes(DayCnt & "Day" ).OnAction = "" & ActiveWorkbook.Name & "!DayBtn" &
191             DayCnt 'Assign Macro
192             DayCnt = DayCnt + 1
193         Next ColCnt
194     Next RowCnt
195
196 'Assign Color Macros
197 For CalCol = 1 To 9
198     .Shapes("CalCol" & CalCol).OnAction = "" & ActiveWorkbook.Name & "!CalCol" &
199     CalCol
200 Next CalCol
201 End With
202
203 With Sheets("CalPopUp" )
204     .Paste
205     .Visible = xlSheetHidden
206
207 'Add in Formulas and Details
208 .Range("A4" ).Value = "=INDEX(CalMonths,A3,)"
209 .Range("A5" ).Value = "=A4&" & Chr(34) & " " & Chr(34) & "&CalYear"
210 .Range("A6" ).Value = "=SUM(B6:H6)"
211 .Range("A8" ).Value = "January"
212 .Range("A8" ).AutoFill Destination:=.Range("A8:A19" ), Type:=xlFillDefault
213 .Range("A20" ).Value =
214 "=IFERROR(INDIRECT(ADDRESS(SUMPRODUCT((B1:H6=A1)*ROW(B1:H6))+6,SUMPRODUCT((B1:H6=A1)*COL
215 UMN(B1:H6)),1,1)),)" & Chr(34) & Chr(34) & ""
216 'Set Defined Names
217 ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
218 ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
219
220 'Add in Calendar Formulas
221 .Range("B1" ).Value =
222 "=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,CalMonth
223 s,0),1)," & Chr(34) & Chr(34) & ""
224 .Range("C1" ).Value = "=IF(B1<>" & Chr(34) & Chr(34) &
225 ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A$4,Cal
226 Months,0),1)," & Chr(34) & Chr(34) & ""
227 .Range("D1" ).Value = "=IF(C1<>" & Chr(34) & Chr(34) &
228 ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A$4,Cal
229 Months,0),1)," & Chr(34) & Chr(34) & ""
230 .Range("E1" ).Value = "=IF(D1<>" & Chr(34) & Chr(34) &
231 ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A$4,Cal
232 Months,0),1)," & Chr(34) & Chr(34) & ""
233 .Range("F1" ).Value = "=IF(E1<>" & Chr(34) & Chr(34) &
234 ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A$4,Cal

```

1 2

```

1 2
221 .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$4,Cal
Months,0),1)," & Chr(34) & Chr(34) & ")"
222 .Range("H1").Value = "=IF(G1<>" & Chr(34) & Chr(34) & "
    ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A$4,Cal
Months,0),1)," & Chr(34) & Chr(34) & ")"
223 .Range("B2").Value = "=H1+1"
224 .Range("C2").Value = "=B2+1"
225 .Range("C2").AutoFill Destination:=.Range("C2:H2"), Type:=xlFillDefault
226 .Range("B2:H2").AutoFill Destination:=.Range("B2:H4"), Type:=xlFillDefault
227 .Range("B5").Value = "=IF(OR(H4=" & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3)," & Chr
(34) & Chr(34) & ",H4+1)"
228 .Range("C5").Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3," & Chr(34) & Chr(34) & "
    ",B5+1)," & Chr(34) & Chr(34) & ")"
229 .Range("B6").Value = "=IFERROR(IF(OR(H5=" & Chr(34) & Chr(34) & "
    ",MONTH(H5+1)<>$A$3)," & Chr(34) & Chr(34) & ",H5+1)," & Chr(34) & Chr(34) & ")"
230 .Range("C6").Value = "=IFERROR(F(MONTH(I5+1)<>$A$3," & Chr(34) & Chr(34) & ",I5+1),"
    & Chr(34) & Chr(34) & ")"
231 .Range("C5:C6").AutoFill Destination:=.Range("C5:H6"), Type:=xlFillDefault
232
233 'Set format to Single Day
234 .Range("B1:H6").NumberFormat = "d"
235
236 'Add in relative Day #'s
237 .Range("B7").Value = "1"
238 .Range("C7").Value = "2"
239 .Range("B8").Value = "8"
240 .Range("C8").Value = "9"
241 .Range("B7:C8").AutoFill Destination:=.Range("B7:H8"), Type:=xlFillDefault
242 .Range("B7:H8").AutoFill Destination:=.Range("B7:H12"), Type:=xlFillDefault
243 GroupCal
244 End With
245 Exit Sub
246 NoCal:
247 MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the calendar
    over from another sheet and paste into this sheet"
248 End Sub
249
250 .....
251 ''''Individual Day Selection Buttons
252 .....
253
254 Sub DayBtn1()
255 With Sheets("CalPopUp")
256 If .Range("B1").Value <> Empty And .Range("A7").Value <> Empty Then
257     ActiveSheet.Range(.Range("A7").Value).Value = .Range("B1").Value
258 End If
259 ActiveSheet.Shapes("Calendar").Visible = msoFalse
260 ActiveCell.Offset(0, 1).Select
261 End With
262 End Sub
263 Sub DayBtn10()
264 With Sheets("CalPopUp")
265 If .Range("D2").Value <> Empty And .Range("A7").Value <> Empty Then
266     ActiveSheet.Range(.Range("A7").Value).Value = .Range("D2").Value
267 End If
268 ActiveSheet.Shapes("Calendar").Visible = msoFalse
269 ActiveCell.Offset(0, 1).Select
270 End With
271 End Sub

```

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

1 2

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

1 2

```
1 2
394 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
395 ActiveCell.Offset(0, 1).Select
396 End With
397 End Sub
398 Sub DayBtn24()
399 With Sheets("CalPopUp" )
400 If .Range("D4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
401 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D4" ).Value
402 End If
403 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
404 ActiveCell.Offset(0, 1).Select
405 End With
406 End Sub
407 Sub DayBtn25()
408 With Sheets("CalPopUp" )
409 If .Range("E4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
410 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E4" ).Value
411 End If
412 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
413 ActiveCell.Offset(0, 1).Select
414 End With
415 End Sub
416 Sub DayBtn26()
417 With Sheets("CalPopUp" )
418 If .Range("F4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
419 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F4" ).Value
420 End If
421 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
422 ActiveCell.Offset(0, 1).Select
423 End With
424 End Sub
425 Sub DayBtn27()
426 With Sheets("CalPopUp" )
427 If .Range("G4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
428 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G4" ).Value
429 End If
430 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
431 ActiveCell.Offset(0, 1).Select
432 End With
433 End Sub
434 Sub DayBtn28()
435 With Sheets("CalPopUp" )
436 If .Range("H4" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
437 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H4" ).Value
438 End If
439 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
440 ActiveCell.Offset(0, 1).Select
441 End With
442 End Sub
443 Sub DayBtn29()
444 With Sheets("CalPopUp" )
445 If .Range("B5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
446 ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B5" ).Value
447 End If
448 ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
449 ActiveCell.Offset(0, 1).Select
450 End With
451 End Sub
452 Sub DayBtn3()
453 With Sheets("CalPopUp" )
454 If .Range("D1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
```

1 2 3


```
1 2 3
455     ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D1" ).Value
456     End If
457     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
458     ActiveCell.Offset(0, 1).Select
459     End With
460 End Sub
461 Sub DayBtn30()
462     With Sheets("CalPopUp" )
463     If .Range("C5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
464         ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C5" ).Value
465     End If
466     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
467     ActiveCell.Offset(0, 1).Select
468     End With
469 End Sub
470 Sub DayBtn31()
471     With Sheets("CalPopUp" )
472     If .Range("D5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
473         ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("D5" ).Value
474     End If
475     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
476     ActiveCell.Offset(0, 1).Select
477     End With
478 End Sub
479 Sub DayBtn32()
480     With Sheets("CalPopUp" )
481     If .Range("E5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
482         ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("E5" ).Value
483     End If
484     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
485     ActiveCell.Offset(0, 1).Select
486     End With
487 End Sub
488 Sub DayBtn33()
489     With Sheets("CalPopUp" )
490     If .Range("F5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
491         ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F5" ).Value
492     End If
493     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
494     ActiveCell.Offset(0, 1).Select
495     End With
496 End Sub
497 Sub DayBtn34()
498     With Sheets("CalPopUp" )
499     If .Range("G5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
500         ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G5" ).Value
501     End If
502     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
503     ActiveCell.Offset(0, 1).Select
504     End With
505 End Sub
506 Sub DayBtn35()
507     With Sheets("CalPopUp" )
508     If .Range("H5" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
509         ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H5" ).Value
510     End If
511     ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
512     ActiveCell.Offset(0, 1).Select
513     End With
514 End Sub
515 Sub DayBtn36()
1
```

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

1

```
1
577 End Sub
578 Sub DayBtn42()
579     With Sheets("CalPopUp" )
580         If .Range("H6" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
581             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H6" ).Value
582         End If
583         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
584         ActiveCell.Offset(0, 1).Select
585     End With
586 End Sub
587 Sub DayBtn5()
588     With Sheets("CalPopUp" )
589         If .Range("F1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
590             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("F1" ).Value
591         End If
592         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
593         ActiveCell.Offset(0, 1).Select
594     End With
595 End Sub
596 Sub DayBtn6()
597     With Sheets("CalPopUp" )
598         If .Range("G1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
599             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("G1" ).Value
600         End If
601         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
602         ActiveCell.Offset(0, 1).Select
603     End With
604 End Sub
605 Sub DayBtn7()
606     With Sheets("CalPopUp" )
607         If .Range("H1" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
608             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("H1" ).Value
609         End If
610         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
611         ActiveCell.Offset(0, 1).Select
612     End With
613 End Sub
614 Sub DayBtn8()
615     With Sheets("CalPopUp" )
616         If .Range("B2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
617             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("B2" ).Value
618         End If
619         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
620         ActiveCell.Offset(0, 1).Select
621     End With
622 End Sub
623 Sub DayBtn9()
624     With Sheets("CalPopUp" )
625         If .Range("C2" ).Value <> Empty And .Range("A7" ).Value <> Empty Then
626             ActiveSheet.Range(.Range("A7" ).Value).Value = .Range("C2" ).Value
627         End If
628         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
629         ActiveCell.Offset(0, 1).Select
630     End With
631 End Sub
632 Sub GroupCal()
633     ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
634     Selection.ShapeRange.Name = "NextMonth"
635     ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
636     Selection.ShapeRange.Name = "PrevMonth"
637     ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
```

1

```

1
638     , "42Day" , "37Day" , "36Day" , "CalBack" , "MonthYear" , "CalBorder" , "1Day" , _
639     "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
640     "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" ,
    "18Day" _
641     , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
642     "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" , "Sa" ,
    _
643     "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" ,
    "CalCol3" , _
644     "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth" ,
    _
645     "NextMonth" )).Visible = msoCTrue
646     ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
647     , "42Day" , "37Day" , "36Day" , "CalBack" , "MonthYear" , "CalBorder" , "1Day" , _
648     "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
649     "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" , "18Day" _
650     , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
651     "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" , "Sa" , _
652     "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" , "CalCol3" , _
    "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth" , _
653     "NextMonth" )).Select
654     Selection.ShapeRange.Group.Select
655     Selection.ShapeRange.Name = "Calendar"
656     Selection.Name = "Calendar"
657     Selection.Placement = xlMove
658     ActiveSheet.Shapes("Calendar" ).Placement = 2
659 End Sub
660
661
662
663
664
665
666
667
668
669 Sub MacroLinkRemover()
670     'PURPOSE: Remove an external workbook reference from all shapes triggering macros
671     'Source: www.ExcelForFreelancers.com
672     Dim shp As Shape
673     Dim MacroLink, NewLink As String
674     Dim SplitLink As Variant
675
676     For Each shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
677
678         'Grab current macro link (if available)
679         On Error GoTo NextShp
680         MacroLink = shp.OnAction
681
682         'Determine if shape was linking to a macro
683         If MacroLink <> "" And InStr(MacroLink, "!") <> 0 Then
684             'Split Macro Link at the exclamation mark (store in Array)
685             SplitLink = Split(MacroLink, "!")
686
687             'Pull text occurring after exclamation mark
688             NewLink = SplitLink(1)
689
690             'Remove any straggling apostrophes from workbook name
691             If Right(NewLink, 1) = "'" Then
692                 NewLink = Left(NewLink, Len(NewLink) - 1)
693             End If

```

1 2 3

```

1 2 3
694
695     'Apply New Link
696     shp.OnAction = NewLink
697     End If
698     NextShp:
699     Next shp
700 End Sub
701 Sub NextMonth()
702     'Next Month button
703     If Sheets("CalPopUp").Range("A20").Value <> Empty Then
704         DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
705         With ActiveSheet.Shapes(DayName)
706             .Fill.ForeColor.RGB = RGB(255, 255, 255)
707             .TextFrame2.TextRange.Font.Bold = msoFalse
708         End With
709     End If
710     With Sheets("CalPopUp")
711         If .Range("A3").Value = 12 Then
712             .Range("A3").Value = 1
713             .Range("A2").Value = .Range("A2").Value + 1
714         Else:
715             .Range("A3").Value = .Range("A3").Value + 1
716         End If
717         If .Range("A6").Value > 0 Then
718             ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day",
719                 "41Day", "42Day")).Visible = True
720         Else:
721             ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day",
722                 "41Day", "42Day")).Visible = False
723         End If
724     End With
725 End Sub
726 Sub PrevMonth()
727     'Previous Month Button
728     If Sheets("CalPopUp").Range("A20").Value <> Empty Then
729         DayName = Sheets("CalPopUp").Range("A20").Value & "Day"
730         With ActiveSheet.Shapes(DayName)
731             .Fill.ForeColor.RGB = RGB(255, 255, 255)
732             .TextFrame2.TextRange.Font.Bold = msoFalse
733         End With
734     End If
735     With Sheets("CalPopUp")
736         If .Range("A3").Value = 1 Then
737             .Range("A3").Value = 12
738             .Range("A2").Value = .Range("A2").Value - 1
739         Else:
740             .Range("A3").Value = .Range("A3").Value - 1
741         End If
742         If .Range("A6").Value > 0 Then
743             ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day",
744                 "41Day", "42Day")).Visible = True
745         Else:
746             ActiveSheet.Shapes.Range(Array("36Day", "37Day", "38Day", "39Day", "40Day",
747                 "41Day", "42Day")).Visible = False
748         End If
749     End With
750 End Sub
751 Sub ShowSettings()
752     'Show or Hide Calendar Settings Panel
753     If ActiveSheet.Shapes.Range(Array("Settings")).Visible = True Then
754         ActiveSheet.Shapes.Range(Array("Settings", "CalCol1", "CalCol2", "CalCol3",

```

1 2

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

—, 88, 89

A

Activate, 80
 ActiveCell, 80, 82-88
 ActiveSheet, 78-91
 ActiveWorkbook, 80, 81
 ActSht, 80
 Add, 80, 81
 Address, 80, 81
 After, 80
 Array, 78-80, 88-91
 AutoFill, 81, 82

B

Bold, 79, 90

C

CalCol, 80, 81
 CalCol1, 78
 CalCol2, 78
 CalCol3, 78
 CalCol4, 78
 CalCol5, 78
 CalCol6, 78
 CalCol7, 78
 CalCol8, 78
 CalCol9, 79
 CalendarHide, 79
 CalendarShow, 79
 CalFormulaReplacement, 79, 80
 Cells, 80, 81
 CheckForSheet, 80
 Chr, 81, 82
 ColCnt, 80, 81
 ColNum, 80
 Copy, 80
 Count, 80
 CreateCalSht, 80
 CreateWS, 80

D

DayBtn1, 82
 DayBtn10, 82
 DayBtn11, 83
 DayBtn12, 83
 DayBtn13, 83
 DayBtn14, 83
 DayBtn15, 83
 DayBtn16, 83
 DayBtn17, 83
 DayBtn18, 84
 DayBtn19, 84
 DayBtn2, 84
 DayBtn20, 84
 DayBtn21, 84
 DayBtn22, 84
 DayBtn23, 84
 DayBtn24, 85
 DayBtn25, 85
 DayBtn26, 85
 DayBtn27, 85
 DayBtn28, 85
 DayBtn29, 85
 DayBtn3, 85
 DayBtn30, 86
 DayBtn31, 86
 DayBtn32, 86

DayBtn33, 86
 DayBtn34, 86
 DayBtn35, 86
 DayBtn36, 86
 DayBtn37, 87
 DayBtn38, 87
 DayBtn39, 87
 DayBtn4, 87
 DayBtn40, 87
 DayBtn41, 87
 DayBtn42, 88
 DayBtn5, 88
 DayBtn6, 88
 DayBtn7, 88
 DayBtn8, 88
 DayBtn9, 88
 DayCnt, 80, 81
 DayName, 78, 79, 90
 DayNum, 79, 80
 Destination, 81, 82
 DrawingObject, 79-81

E

Empty, 79, 82-88, 90

F

Fill, 78, 79, 90
 Font, 79, 90
 ForeColor, 78, 79, 90
 Formula, 79-81

G

Group, 88, 89
 GroupCal, 79, 82, 88

I

InStr, 79, 89
 IsDate, 79

L

Left, 80, 89
 Len, 89

M

MacroLink, 89
 MacroLinkRemover, 79, 89
 Month, 79
 MsgBox, 80, 82
 msoCTrue, 80, 89
 msoFalse, 79, 82-88, 90
 msoTrue, 79

N

Name, 80, 81, 88, 89
 Names, 81
 NewLink, 89, 90
 NextMonth, 90
 NextShp, 89, 90
 NoCal, 79, 80, 82
 NumberFormat, 82

O

Offset, 80, 82-88
 OnAction, 79, 81, 89, 90

P

Paste, 81
 Placement, 89
 PrevMonth, 90

R

Range, 78-91
 RefersTo, 81
 RGB, 78, 79, 90
 Right, 89
 RowCnt, 80, 81
 RowNum, 80

S

SelCell, 78-80
 Selection, 78, 79, 88, 89
 Shape, 80, 89
 ShapeRange, 78, 79, 88, 89
 Shapes, 78-91
 Sheets, 78-88, 90
 ShowSettings, 90
 shp, 80, 89, 90
 Split, 89
 SplitLink, 89

T

TextFrame2, 79, 90
 TextRange, 79, 90
 ThisWorkbook, 80
 Top, 80

U

Ungroup, 91
 UnGroupCal, 79, 81, 91

V

Value, 78-88, 90
 Visible, 79-91

W

Worksheet, 80
 ws, 80

X

xlFillDefault, 81, 82
 xlMove, 89
 xlSheetHidden, 81

Y

Year, 79

```
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     With Sheet4
103         LastLeaveTypeRow = .Range("G24" ).End(xlUp).Row
104         On Error Resume Next
105         With Application
106             .CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
107             .CommandBars("Cell" ).Controls("Go To Event" ).Delete
108             Set LeaveBtn = .CommandBars("Cell" ).Controls.Add(Before:=1, Type:=msoControlPopup)
109             Set GoToEventBtn = .CommandBars("Cell" ).Controls.Add(Before:=2, Type:=
110                 msoControlButton)
111         End With
112         With LeaveBtn
113             .Caption = "Add Leave Event"
114             For LeaveTypeRow = 7 To LastLeaveTypeRow
115                 With .Controls.Add(Type:=msoControlButton)
116                     .Caption = Sheet4.Range("G" & LeaveTypeRow).Value
117                     .FaceId = 612
118                     .OnAction = "CreateLeaveEvent" & LeaveTypeRow
119                 End With
120             Next LeaveTypeRow

```

1 2 3

```
120     End With
121     With GoToEventBtn
122         .Caption = "Go To Event"
123         .FaceId = 1812
124         .OnAction = "Sched_GoToEvent"
125     End With
126 End With
127 End Sub
128 Sub RightClick_Cancel()
129     On Error Resume Next
130     With Application
131         .CommandBars("Cell" ).Controls("Add Leave Event" ).Delete
132         .CommandBars("Cell" ).Controls("Go To Event" ).Delete
133     End With
134     On Error GoTo 0
135 End Sub
```

A

Activate, 93
ActiveCell, 93
Add, 94
Application, 94, 95

B

Before, 94

C

Caption, 94, 95
Cells, 93
Column, 93
Columns, 93
CommandBarButton, 94
CommandBarPopup, 94
CommandBars, 94, 95
Controls, 94, 95
Count, 93
CreateLeaveEvent, 93, 94
CreateLeaveEvent10, 93
CreateLeaveEvent11, 93
CreateLeaveEvent12, 93
CreateLeaveEvent13, 93
CreateLeaveEvent14, 93
CreateLeaveEvent15, 93
CreateLeaveEvent16, 93
CreateLeaveEvent17, 93
CreateLeaveEvent18, 93
CreateLeaveEvent19, 94
CreateLeaveEvent20, 94
CreateLeaveEvent21, 94
CreateLeaveEvent22, 94
CreateLeaveEvent23, 94
CreateLeaveEvent7, 94
CreateLeaveEvent8, 94
CreateLeaveEvent9, 94

D

DateRow, 93
Delete, 94, 95

E

EmpRow, 93
Event_AddNew, 93
Explicit, 93

F

FacelId, 94, 95

G

GoToEventBtn, 94, 95

L

LastLeaveTypeRow, 94
LeaveBtn, 94
LeaveTypeRow, 94

M

msoControlButton, 94
msoControlPopup, 94

O

OnAction, 94, 95

R

Range, 93, 94
RightClick_CallEvents, 94
RightClick_Cancel, 95

Row, 93, 94

S

SelCol, 93
SelColumns, 93
Selection, 93
SelLeaveRow, 93, 94
SelRow, 93
Sheet2, 93
Sheet4, 93, 94
Sheet5, 93
Sheet8, 93

V

Value, 93, 94

X

xLUp, 94

```

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

```

1 2 3

```

56   1 2 3 End If
57   ActiveCell.Comment.Visible = True
58   End With
59 End Sub
60
61 Sub Sched_AddComment_Holiday()
62     Dim HolidayDt As Date
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         HolidayName = Sheet4.Range("D" & HolidayRow).Value
74         ActiveCell.ClearComments
75         With ActiveCell.AddComment
76             With ActiveCell.Comment
77                 .Shape.Width = 200
78                 .Shape.Height = 20
79                 .Shape.AutoShapeType = msoShapeRoundedRectangle
80                 .Shape.TextFrame.Characters.Font.Name = "Tahoma"
81                 .Shape.TextFrame.Characters.Font.Size = 11
82                 .Shape.TextFrame.Characters.Font.Bold = False
83                 .Shape.TextFrame.Characters.Font.ColorIndex = 2
84                 .Shape.Line.ForeColor.RGB = RGB(0, 0, 0)
85                 .Shape.Line.BackColor.RGB = RGB(255, 255, 255)
86                 .Shape.Fill.Visible = msoTrue
87                 .Shape.Fill.ForeColor.RGB = RGB(102, 0, 204)
88                 .Shape.Fill.OneColorGradient msoGradientDiagonalUp, 1, 0.23
89                 .text text:="Holiday: " & HolidayName
90             End With
91         End With
92     End If
93 End Sub
94 Sub Sched_ClearFilters()
95     With Sheet8
96         .Range("DD3,T3" ).Value = "[All Types/Positions]"
97         .Range("DF3,AC3" ).Value = "[All Event Types]"
98         If ActiveSheet.AutoFilterMode Then
99             .AutoFilter.ShowAllData
100            .AutoFilterMode = False
101        End If
102    End With
103 End Sub
104 Sub Sched_GoToEvent()
105     With Sheet8
106         Dim ActRow As Long
107         Dim ActCol As Long
108         Dim EventID As Long
109         ActRow = ActiveCell.Row
110         ActCol = ActiveCell.Column
111         If .Cells(ActRow, ActCol + 60).Value = Empty Then
112             MsgBox "Please select a valid event to open"
113             Exit Sub
114         End If
115         EventID = .Cells(ActRow, ActCol + 60).Value 'Event ID
116         Sheet5.Activate

```

1 2

```

1 2
117 Sheet5.Range("B4" ).Value = EventID
118 Event_Load 'Run Load Macro Event
119 End With
120 End Sub
121
122 Sub Sched_MonthView()
123 With Sheet8
124 StopCalc
125 .Shapes("MonthGrp" ).Visible = msoCTrue
126 .Shapes("WeekGrp" ).Visible = msoFalse
127 .Range("F:CZ" ).EntireColumn.Hidden = False
128 .Range("DA:F0" ).EntireColumn.Hidden = True
129 ActiveWindow.FreezePanes = False
130 Range("F8" ).Select
131 ActiveWindow.FreezePanes = True
132
133 ResetCalc
134 End With
135 End Sub
136
137 Sub Sched_NextMonth()
138 Sched_ClearFilters
139 With Sheet8
140 .Range("F8:AK314" ).ClearComments 'Clear All Comments
141 If .Range("B1" ).Value = 12 Then
142 .Range("E2" ).Value = .Range("E2" ).Value + 1
143 .Range("B1" ).Value = 1
144 Else: .Range("B1" ).Value = .Range("B1" ).Value + 1
145 End If
146 End With
147 Sched_Refresh
148 End Sub
149
150 Sub Sched_NextWeek()
151 Sched_ClearFilters
152 With Sheet8
153 .Range("B2" ).Value = .Range("B2" ).Value + 7
154 If Year(.Range("B2" ).Value) <> .Range("E2" ).Value Then .Range("E2" ).Value = .Range(
155 "E2" ).Value + 1
156 .Range("F8:DG314" ).ClearComments 'Clear All Comments
157 Sched_Refresh
158 End With
159 End Sub
160
161 Sub Sched_PrevMonth()
162 Sched_ClearFilters
163 With Sheet8
164 .Range("F8:AK314" ).ClearComments 'Clear All Comments
165 If .Range("B1" ).Value = 1 Then
166 .Range("E2" ).Value = .Range("E2" ).Value - 1
167 .Range("B1" ).Value = 12
168 Else: .Range("B1" ).Value = .Range("B1" ).Value - 1
169 End If
170 End With
171 Sched_Refresh
172 End Sub
173
174 Sub Sched_PrevWeek()
175 With Sheet8
176 .Range("B2" ).Value = .Range("B2" ).Value - 7
177 If Year(.Range("B2" ).Value) <> .Range("E2" ).Value Then .Range("E2" ).Value = .Range(

```

1 2

```
1 2
177     "E2" ).Value - 1
178     .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
201     Else: Sheet8.Range("7:7" ).EntireRow.Hidden = True
202 End Sub
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 Sub Sched_ThisWeek()
210     Sched_ClearFilters
211     Sheet8.Range("B2" ).Value = Sheet8.Range("A11" ).Value
212     Sched_Refresh
213 End Sub
214 Sub Sched_WeekView()
215     With Sheet8
216         StopCalc
217         .Shapes("MonthGrp" ).Visible = msoFalse
218         .Shapes("WeekGrp" ).Visible = msoCTrue
219         .Range("F:CZ" ).EntireColumn.Hidden = True
220         .Range("DA:F0" ).EntireColumn.Hidden = False
221         ActiveWindow.FreezePanes = False
222         Range("DA8" ).Select
223         ActiveWindow.FreezePanes = True
224         .Range("F8:DG309" ).ClearComments 'Clear All Comments
225         ResetCalc
226     End With
227 End Sub
```

→, 97

A

ActCol, 97, 98
 Activate, 98
 ActiveCell, 97, 98
 ActiveSheet, 98
 ActiveWindow, 99, 100
 ActRow, 97, 98
 AddComment, 97, 98
 Application, 100
 AutoFilter, 98
 AutoFilterMode, 98
 AutoShapeType, 97, 98

B

BackColor, 97, 98
 Bold, 97, 98

C

Calculate, 100
 Cells, 97, 98
 Characters, 97, 98
 ClearComments, 97-100
 ColorIndex, 97, 98
 Column, 97, 98
 Comment, 97, 98
 Copy, 100
 CutCopyMode, 100

E

Empty, 97, 98
 EndOn, 97
 EntireColumn, 99, 100
 EntireRow, 100
 Event_Load, 99
 EventComment, 97
 EventCreatedOn, 97
 EventEmpl, 97
 EventID, 97-99
 EventIDRng, 97
 EventName, 97
 EventNotes, 97
 EventRecFreq, 97
 EventRecQty, 97
 EventRecur, 97
 EventRow, 97
 EventType, 97
 Explicit, 97

F

Fill, 97, 98
 Find, 97, 98
 Font, 97, 98
 ForeColor, 97, 98
 FoundEventIDRng, 97
 FoundHolDtRng, 98
 FreezePanels, 99, 100

H

Height, 97, 98
 Hidden, 99, 100
 HolDtRng, 98
 HolidayDt, 98
 HolidayName, 98
 HolidayRow, 98

L

Line, 97, 98

M

Month, 100
 MsgBox, 98
 msoCTrue, 99, 100
 msoFalse, 99, 100
 msoGradientDiagonalUp, 97, 98
 msoShapeRoundedRectangle, 97, 98
 msoTrue, 97, 98

N

Name, 97, 98
 Now, 100

O

OneColorGradient, 97, 98

P

PasteSpecial, 100

R

Range, 97-100
 ResetCalc, 99, 100
 RGB, 97, 98
 Row, 97, 98

S

Sched_AddComment_Event, 97
 Sched_AddComment_Holiday, 98
 Sched_ClearFilters, 98-100
 Sched_GoToEvent, 98
 Sched_MonthView, 99
 Sched_NextMonth, 99
 Sched_NextWeek, 99
 Sched_PrevMonth, 99
 Sched_PrevWeek, 99
 Sched_Refresh, 99, 100
 Sched_ScrollToRow, 100
 Sched_ShowHideCapacity, 100
 Sched_ThisMonth, 100
 Sched_ThisWeek, 100
 Sched_WeekView, 100
 ScrollRow, 100
 Shape, 97, 98
 Shapes, 99, 100
 Sheet4, 98
 Sheet5, 98, 99
 Sheet7, 97
 Sheet8, 97-100
 ShowAllData, 98
 Size, 97, 98
 StartOn, 97
 StopCalc, 99, 100

T

text, 97, 98
 TextFrame, 97, 98

V

Value, 97-100
 vbCrLf, 97
 Visible, 97-100

W

Width, 97, 98

X

xlPasteFormulas, 100
 xlPasteValues, 100
 xlValues, 97, 98
 xlWhole, 97, 98

Y

Year, 99, 100


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

1

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

1 2

```

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

```

1 2

```

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

```

1

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

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

1

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

1

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



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

```

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

```

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

—, 105, 106

A

ActiveCell, 102, 103, 111
 ActiveSheet, 102-112
 AMPMSel, 102
 AMSel, 102
 Application, 104, 105
 Array, 102, 103, 105, 110, 111

C

Characters, 102-109, 111
 CurLeft, 103-105
 CurTop, 103-105

D

DoneBtn, 102

E

Empty, 103, 111
 ExitSub, 102, 103, 105, 106
 Explicit, 102

F

Fill, 103, 105, 109-111
 Font, 103, 105
 ForeColor, 103, 105, 109-111
 Format, 111

G

Group, 102, 103, 106
 GrpHrs, 102, 111
 GrpMin, 102, 111
 GrpRange, 102, 103, 105, 106
 GrpSet, 102, 111
 GrpShapes, 102, 103, 105, 106

H

Height, 104, 105
 HideHrSel, 103, 105-108, 111
 HideMinSel, 103, 108, 109, 111
 HoursDisplay, 103, 111
 Hr, 111

I

InStr, 104, 111
 Inter, 104, 105
 Intg10, 104, 105
 Intg15, 104, 105
 Intg5, 104, 105

L

Left, 104, 105, 111
 Len, 104

M

MacroLink, 104
 MacroLinkRemover, 104, 111
 MinDisplay, 104, 106-108, 110
 MsgBox, 111
 msoCTrue, 109-111
 msoFalse, 110, 111

N

Name, 102, 103, 106
 NewLink, 104
 NextShp, 104
 NoTP, 111

Now, 104, 105

O

Offset, 111
 OnAction, 104, 111

P

Placement, 106
 PMSel, 102, 105
 ProtTP, 102, 104-111

R

Range, 102, 103, 105, 110, 111
 RGB, 103, 105, 109-111
 Right, 104, 111

S

SelCell, 111
 Select1, 106
 Select10, 106
 Select11, 106
 Select12, 106
 Select2, 106
 Select3, 107
 Select4, 107
 Select5, 107
 Select6, 107
 Select7, 107
 Select8, 107
 Select9, 107
 Selection, 111
 SelectM00, 108, 110
 SelectM05, 108
 SelectM10, 108
 SelectM15, 108
 SelectM20, 108
 SelectM25, 108
 SelectM30, 109
 SelectM35, 109
 SelectM40, 109
 SelectM45, 109
 SelectM50, 109
 SelectM55, 109
 Set10MinInt, 109
 Set15MinInt, 110
 Set5MinInt, 110
 SetHideShow, 110
 SetThm1, 110
 SetThm2, 110
 SetThm3, 110
 SetThm4, 110
 SetThm5, 111
 Shape, 104
 Shapes, 102-112
 shp, 104
 Siz, 103-105
 Split, 104
 SplitLink, 104

T

text, 102-109, 111
 TextFrame, 102-109, 111
 TextFrame2, 103, 105
 TextRange, 103, 105
 Top, 104, 105, 111
 TPShow, 111

U

Ungroup, 111, 112
 UnGrpHrs, 111

UnGrpMin, 111, 112
 UnGrpSet, 111, 112
 UnProtTP, 102-112

V

Value, 102, 103, 111
 Visible, 102-111

W

Wait, 104, 105
 Width, 104, 105

```
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
```

A

ActiveSheet, 114, 117
Application, 115, 117

B

BreakThreshold, 115, 116

C

ClearContents, 115, 117

D

Day, 115
DayThreshold, 115, 116
delay, 114, 117, 118
Delete, 114
DoEvents, 114, 118

E

EmpID, 114-116
EmpName, 114-117
EmpRow, 114, 115, 117
Empty, 115-117
EntryAdded, 115-117
EntryAddedAlready, 117
EntryType, 114-117
Explicit, 114

F

Fill, 114, 118
Find, 115
FoundPrevTimeClockIDRng, 115
FoundTimeClockIDRng, 114-116

H

Height, 114

I

i, 114, 117, 118
IncrementLeft, 114
IncrementTop, 114
Insert, 114

L

Left, 114
LockAspectRatio, 114

M

MessRow, 114, 117
Month, 115
msoCTrue, 117
msoFalse, 115, 117
msoSoftEdgeType4, 114
msoTrue, 114

N

Name, 114
NoPic, 114
Now, 115-118

P

PicPath, 114
Pictures, 114

R

Range, 114-118
Replace, 117
Round, 117
Row, 115, 116
RunClk, 114, 117, 118

S

ScreenUpdating, 117
ShapeRange, 114
Shapes, 114, 115, 117
Sheet12, 114, 115, 117, 118
Sheet2, 114, 117
Sheet4, 115-117
Sheet9, 115, 117
SoftEdge, 114
StartTime, 114, 117, 118

T

text, 115
Timeclock_AlreadyEntered, 114, 117
Timeclock_DisplayPicture, 114, 117
Timeclock_EnterID, 114
Timeclock_IDNotFound, 115, 117
Timeclock_StartStop, 118
TimeClockID, 114-116
TimeClockID_Prev, 114, 115
TimeClockIDRng, 114, 115
TimeClockRow, 114-117
Timer, 114, 118
TimeRow, 114, 117
TimeValue, 117, 118
Top, 114
TotalBreakHours, 115, 117
TotalHours, 115, 117
Transparency, 114, 118

V

Value, 114-117
Visible, 114, 115, 117, 118

W

Wait, 117
WorksheetFunction, 115

X

xLUp, 116
xlValues, 115
xlWhole, 115

Y

Year, 115

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, [120](#)
Application, [120](#)

B

BreakIn, [120](#), [121](#)
BreakOut, [120](#), [121](#)
BrowseRemoteTimeClockFolder, [120](#)

C

ClockIn, [120](#), [121](#)
ClockOut, [120](#), [121](#)
Close, [120](#)
createfolder, [120](#)
CreateObject, [120](#)

D

Destination, [121](#)
Dir, [120](#), [121](#)

E

EmpID, [120](#), [121](#)
EmpIDRng, [120](#), [121](#)
Empty, [120](#), [121](#)
Explicit, [120](#)

F

FileName, [120](#), [121](#)
Find, [120](#), [121](#)
FoundEmpIDRng, [120](#), [121](#)
FoundTimeClockIDRng, [120](#)
FSO, [120](#), [121](#)

I

Input, [120](#)
InStr, [120](#)

L

Len, [120](#)
Line, [120](#)

M

Mid, [120](#)
MoveFile, [121](#)
MsgBox, [120](#)

O

Open, [120](#)

R

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

S

Sheet2, [120](#), [121](#)
Sheet4, [120](#)
Sheet9, [120](#)
Source, [121](#)

T

TCTFile, [120](#), [121](#)
TCText, [120](#)
TCTextLine, [120](#)
text, [120](#)
TimeClock_SyncRemoteEntries, [120](#)
TimeClockFolder, [120](#), [121](#)
TimeClockID, [120](#), [121](#)
TimeClockIDRng, [120](#)
TimeClockRow, [120](#), [121](#)

V

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

W

WorksheetFunction, [120](#)

X

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

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