VBA Code Documentation – Open\_Close\_Procedure – RCL Library Change Document

These procedures set up the workbook so that it is more user friendly and processes information faster.

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'This is needed for the TimeToMillisecond Function below

Private Type SYSTEMTIME

wYear As Integer

wMonth As Integer

wDayOfWeek As Integer

wDay As Integer

wHour As Integer

wMinute As Integer

wSecond As Integer

wMilliseconds As Integer

End Type

<http://custom-designed-databases.com/wordpress/2011/get-milliseconds-or-seconds-from-system-time-with-vba/>

Private Declare Sub GetSystemTime Lib "kernel32" \_

(lpSystemTime As SYSTEMTIME)

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Private Sub Auto\_Open() <- Creating the Auto Open Procedure

This function automatically runs every time the workbook is opened (This is referred to as a new session in alot of the commenting).

The purpose of this function is to populate the "Data tab" sheet with information regarding who used the workbook, as well as when they used it.

In the function located in the Email\_Function module, the date and time of when emails were sent is added next to this session information.

Dim Entry As Range <- Assigning variables

Dim mode As String <- Assigning variables

'Call Opening\_Procedure

Call LudicrousMode(False) <- Speeds up worksheet processing

<https://www.reddit.com/r/excel/comments/66o8jh/is_using_excel_the_best_solution_here/>

mode = returnUserMode(Environ$("Username"))

If mode = "SPOC" Or mode = "DEV" Then

Call adminMode

Else

Call baseMode

End If

This is going to grab the username and m0 number to see if the user falls into the category of a SPOC (Single Point of Contact) or is a developer. If they are one of these, then they get special permissions as an admin, and if they aren’t, then they get the base use ability.

Template ID is the datetime stamp from the first time the template workbook is opened from SharePoint.

After the initial opening, the template ID is never updated so it remains consistent through its use.

Call createTemplateID

Saves the user mode to the data tab based on the current user's M0.

Call storeMode

Set sheets to protect userInterfaceOnly = true.

Sheets("Library View").Protect UserInterfaceOnly:=True, Password:="test"

This sets a range variable to the time entry for the new session.

Set Entry = Worksheets("Data Tab").Range("O" & WorksheetFunction.CountA(Worksheets("Data Tab").Range("O:O")) + 2)

The first entry will be the current user's M0 Number. This will be added to the table for tracking in sequential order.

Entry = (Environ$("Username"))

The entry next to the first will be the date and time that the session began

Entry.Offset(0, 1) = Format(Now, "dd-mmm-yy h mm am/pm")

This renames the Email button in the "Library View" sheet. This sub is located in the "Rename\_Email\_Button" module.

Call Name\_Email\_Button

'Call Closing\_Procedure

The sheet "Library View" is activated so it is the sheet that is seen when the workbook is opened.

Worksheets("Library View").Activate

End Sub

The Dim and Set statements are identified as worksheets and then set to the data tab sheet and the data tab index sheet. The data tab sheet holds the data, but the index sheet is dynamic and determines which columns go to which table based on a header.

Sub createTemplateID()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

This grabs the date and puts it into the data tab so that whenever a user is captured and put into the table, you know when they were.

Dim queryPosition As Range

Dim indexRow As Long

Dim numCol As Long

Dim startCol As Long

Dim SQLstr As String

SQLstr = ""

indexRow = WorksheetFunction.Match("Template ID", dataIndexWS.Range("A1:A1000"), 0)

numCol = dataIndexWS.Cells(indexRow, 3)

startCol = dataIndexWS.Cells(indexRow, 2)

If dataWS.Range(Col\_Letter(startCol) & 3) = "" Then

dataWS.Range(Col\_Letter(startCol) & 3) = Format(Date, "YYYY-MM-DD") & "-" & TimeToMillisecond

End If

End Sub

This builds the time value sRet and builds the TimeToMillisecond value.

Public Function TimeToMillisecond() As String

Dim tSystem As SYSTEMTIME

Dim sRet

On Error Resume Next

GetSystemTime tSystem

sRet = Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & \_

":" & tSystem.wMilliseconds

TimeToMillisecond = sRet

End Function

Sub storeMode()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Dim i As Integer

Dim j As Integer

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

i = 1 'stores the row in the index corresponding to the dataset needed

Do While dataIndexWS.Cells(i, 1) <> "" <- When not equal to nothing

If dataIndexWS.Cells(i, 1) = "User Mode" <- Then Loop through until in user mode

Exit Do

End If

i = i + 1 <- If the value equals initial value plus 1

Loop

This selects the next cell and inserts the username

dataWS.Cells(3, dataIndexWS.Cells(i, 2)) = returnUserMode(Environ$("Username"))

End Sub

This thing is really just to check if you’re a SPOC, and if you are, then to add additional options for them to use in facilitating the tool. They will be able to see the additional columns that aren’t as relevant to the everyday user. This is the ADMIN MODE, SON!

Sub adminMode()

Dim objWS As Worksheet

Dim riskWS As Worksheet

Dim contWS As Worksheet

Dim libWS As Worksheet

Dim viewWS As Worksheet

Set objWS = Sheets("Objectives")

Set riskWS = Sheets("Risks")

Set contWS = Sheets("Controls")

Set libWS = Sheets("Library View")

Set viewWS = Sheets("Result View")

objWS.Shapes("Check Box 6").Visible = True ' SPOC unhide

objWS.Shapes("Check Box 6").ControlFormat.Value = 1

' Check SPOC checkbox to unhide SPOC cols

Call objSPOC\_Click

riskWS.Shapes("Check Box 15").Visible = True ' SPOC unhide

riskWS.Shapes("Check Box 15").ControlFormat.Value = 1 ' Check SPOC checkbox to unhide SPOC cols

Call riskSPOC\_Click

contWS.Shapes("Check Box 10").Visible = True ' SPOC unhide

contWS.Shapes("Check Box 10").ControlFormat.Value = 1 ' Check SPOC checkbox to unhide SPOC cols

Call controlSPOC\_Click

libWS.Shapes("Check Box 13").Visible = True ' SPOC unhide

libWS.Shapes("Check Box 13").ControlFormat.Value = 1 ' Check SPOC checkbox to unhide SPOC cols

Call libSPOC\_Click

End Sub

This is for non admins, they can’t see anything they aren’t supposed to. If you’re reading this, you’re probably cool enough to be in the admin club, so maybe go get a sixer of Dr. Peppers and meet us behind the Boston Avenue Grill and we can talk about admin stuff.

Sub baseMode()

Dim objWS As Worksheet

Dim riskWS As Worksheet

Dim contWS As Worksheet

Set objWS = Sheets("Objectives")

Set riskWS = Sheets("Risks")

Set contWS = Sheets("Controls")

objWS.Shapes("Check Box 6").Visible = False ' SPOC hide

objWS.Shapes("Check Box 6").ControlFormat.Value = -4146 ' Uncheck SPOC checkbox to hide SPOC cols

Call objSPOC\_Click

riskWS.Shapes("Check Box 15").Visible = False ' SPOC unhide

riskWS.Shapes("Check Box 15").ControlFormat.Value = -4146 ' Check SPOC checkbox to unhide SPOC cols

Call riskSPOC\_Click

contWS.Shapes("Check Box 10").Visible = False ' SPOC unhide

contWS.Shapes("Check Box 10").ControlFormat.Value = -4146 ' Check SPOC checkbox to unhide SPOC cols

Call controlSPOC\_Click

libWS.Shapes("Check Box 13").Visible = False ' SPOC unhide

libWS.Shapes("Check Box 13").ControlFormat.Value = -4146 ' Check SPOC checkbox to unhide SPOC cols

Call libSPOC\_Click

End Sub

This grabs the users M0 number which will be used to find who their O R C owner is and also build their identity into the email chain… and worksheet tracking.

Or is this the part where the check is being done to see if the M0 number of the user is in the list of approved admin users? Then if they aren’t in that, return the base user profile?

Public Function returnUserMode(M0 As String)

Dim dataWS As Worksheet <- data sheet

Dim dataIndexWS As Worksheet <- Index sheet for data sheet

Dim i As Integer

Dim j As Integer

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Stores the row in the index corresponding to the dataset needed

i = 1

Do While dataIndexWS.Cells(i, 1) <> ""

If dataIndexWS.Cells(i, 1) = "Admin Info" Then

Exit Do

End If

i = i + 1

Loop

Stores the row being reviewed in the Data Tab

j = 1

Do While dataWS.Cells(j, dataIndexWS.Cells(i, 2)) <> ""

'MsgBox dataWS.Cells(j, dataIndexWS.Cells(i, 2) + 1)

If UCase(M0) = dataWS.Cells(j, dataIndexWS.Cells(i, 2) + 1) Then

returnUserMode = dataWS.Cells(j, dataIndexWS.Cells(i, 2) + 3)

Exit Function

End If

j = j + 1

Loop

returnUserMode = "Base User"

End Function

I’m not really sure what this one does.

Function Col\_Letter(lngCol As Long) As String

Dim vArr

vArr = Split(Cells(1, lngCol).Address(True, False), "$")

Col\_Letter = vArr(0)

End Function

Sub LudicrousMode(ByVal Toggle As Boolean, Optional ByVal targetWS As Worksheet)

Dim ws As Worksheet

Dim pwd As String

Application.ScreenUpdating = Not Toggle

' Application.EnableEvents = Not Toggle

' Application.DisplayAlerts = Not Toggle

Application.EnableAnimations = Not Toggle

' Application.DisplayStatusBar = Not Toggle

' Application.PrintCommunication = Not Toggle

Application.Calculation = IIf(Toggle, xlCalculationManual, xlCalculationAutomatic)

<http://www.excelfunctions.net/vba-iif-function.html>

'Unlocks all worksheets if xlCalculculationManual

' pwd = "test" ' Put your password here

' If targetWS Is Nothing Then

' If Toggle Then

' For Each ws In Worksheets

' ws.Unprotect Password:=pwd

' Next ws

' Else

' For Each ws In Worksheets

' ws.Protect Password:=pwd

' Next ws

' End If

' Else

' If Toggle Then

' targetWS.Unprotect Password:=pwd

' Else

' targetWS.Protect Password:=pwd

' End If

' End If

End Sub