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

This is how the excel sheet will pull the data from the SQL Server Database and move it into the data tab using the data index, then grab IDs and Descriptions and put them on their respective worksheets.

First we make sure the data index is in order by deleting all of the past data so that we are only using the most up to date values from the server. Then it loops through and copies over column headers and keeps the A column as a reference from the data sheet. Anything that doesn’t carry over that should have has its name printed in a message box describing the missing parts.

Sub updateDataIndex()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Dim i As Integer

Dim j As Integer

Dim missing() As String

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

dataIndexWS.Range("B2:C1000").Clear

dataIndexWS.Range("E2:AAA1000").Clear

'loop through names in Col A of Data Index tab

' i stores the col A row ref

i = 2

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

If IsError(Application.Match(dataIndexWS.Cells(i, 1).Value, dataWS.Range("A1:AAA1"), 0)) Then

dataIndexWS.Range("A" & i).Interior.Color = 255

MsgBox dataIndexWS.Cells(i, 1) & " was not found in the Data Tab."

Else

dataIndexWS.Cells(i, 2) = WorksheetFunction.Match(dataIndexWS.Cells(i, 1), dataWS.Range("A1:AAA1"), 0)

'loop through dataWS to count number of columns in data range and copy over column headers

'j stores the header col ref

j = 0

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

dataIndexWS.Cells(i, 5 + j) = dataWS.Cells(2, dataIndexWS.Cells(i, 2) + j)

j = j + 1

Loop

dataIndexWS.Cells(i, 3) = j

End If

i = i + 1

Loop

'If a name in Col A of index is not found store and product msgbox warning

'with missing names

End Sub

This pulls over the EMP\_MGR table. This table contains the employee and manager’s names, M0 numbers, and email addresses. This will be for the email chain.

Sub queryEMP\_MGR()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim queryPosition As Range

Dim indexRow As Long

Dim numCol As Long

Dim startCol As Long

Dim SQLstr As String

SQLstr = ""

indexRow = WorksheetFunction.Match("EMP / MGR", dataIndexWS.Range("A1:A1000"), 0)

numCol = dataIndexWS.Cells(indexRow, 3)

startCol = dataIndexWS.Cells(indexRow, 2)

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCol - 1) & 10000).Clear

SQLstr = SQLstr & "SELECT [EMP\_Name],[EMP\_M0],[EMP\_Email]" \_

& ",[MGR\_Name],[MGR\_M0],[MGR\_Email]" \_

& "FROM [ERMDB].[dbo].[EMP\_MGR]"

Call QueryERMDB(SQLstr, dataWS.Range(Col\_Letter(startCol) & 3), False)

End Sub

This table contains the SPOC and Control Owner’s names, M0 numbers, and email addresses. This will be for the email chain.

Sub queryCO\_SPOC()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim queryPosition As Range

Dim indexRow As Long

Dim numCol As Long

Dim startCol As Long

Dim SQLstr As String

SQLstr = ""

indexRow = WorksheetFunction.Match("CO / SPOC", dataIndexWS.Range("A1:A1000"), 0)

numCol = dataIndexWS.Cells(indexRow, 3)

startCol = dataIndexWS.Cells(indexRow, 2)

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCol - 1) & 10000).Clear

SQLstr = SQLstr & "SELECT [CO\_Name],[CO\_M0],[CO\_Email],[SPOC],[SPOC\_M0]" \_

& ",[SPOC\_Email] FROM [ERMDB].[dbo].[CO\_SPOC]"

Call QueryERMDB(SQLstr, dataWS.Range(Col\_Letter(startCol) & 3), False)

End Sub

This grabs all of the columns out of the Objectives Table. For more information please refer to the data dictionary.

Sub queryObjectives()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim queryPosition As Range

Dim indexRow As Long

Dim numCol As Long

Dim startCol As Long

Dim SQLstr As String

SQLstr = ""

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

numCol = dataIndexWS.Cells(indexRow, 3)

startCol = dataIndexWS.Cells(indexRow, 2)

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCol - 1) & 10000).Clear

SQLstr = SQLstr & "SELECT [Objectives\_Primary\_Key],[Objective\_ID],[Objective\_Description]" \_

& " ,[Objective\_Owner],[Objective\_Delegate],[User\_Notes],[User\_Field\_1]" \_

& " ,[User\_Field\_2],[Objective\_Owner\_M0],[Objective\_Owner\_Delegate\_M0]" \_

& " FROM [ERMDB].[dbo].[Objectives\_Test]"

Call QueryERMDB(SQLstr, dataWS.Range(Col\_Letter(startCol) & 3), False)

End Sub

This grabs all of the columns out of the Risks Table. For more information please refer to the data dictionary.

Sub queryRisks()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim queryPosition As Range

Dim indexRow As Long

Dim numCol As Long

Dim startCol As Long

Dim SQLstr As String

SQLstr = ""

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

numCol = dataIndexWS.Cells(indexRow, 3)

startCol = dataIndexWS.Cells(indexRow, 2)

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCol - 1) & 10000).Clear

SQLstr = SQLstr & "SELECT [Risks\_Primary\_Key],[Objective\_ID],[Risk\_ID],[Risk\_Description]" \_

& " ,[Risk\_Owner],[Risk\_Description\_Detail],[Risk\_Owner\_Delegate],[User\_Notes\_R1]" \_

& " ,[Custom\_Risk\_ID],[Residual\_Risk],[Residual\_Risk\_Rationale],[Control\_Environment]" \_

& " ,[Control\_Environment\_Rationale],[Residual\_Risk\_Dollars]" \_

& " ,[Severity\_Risk\_Rating\_Custom],[Likelihood\_Risk\_Rating\_Custom]" \_

& " ,[Inherent\_Risk\_Score\_Custom],[Residual\_Risk\_Custom],[Risk\_Process\_ID]" \_

& " ,[Risk\_Primary\_Cause],[Ops\_Risk\_Subtype\_Cause],[Comp\_Risk\_Subtype\_Cause]" \_

& " ,[Risk\_Primary\_Effect],[Ops\_Risk\_Subtype\_Effect],[Comp\_Risk\_Subtype\_Effect]" \_

& " ,[Reputational\_Flag],[Ops\_Basel\_Cause\_1],[Ops\_Basel\_Effect\_1]" \_

& " ,[Ops\_Basel\_Cause\_2],[Ops\_Basel\_Effect\_2],[Ops\_Basel\_Cause\_3]" \_

& " ,[Ops\_Basel\_Effect\_3],[User\_Field\_1],[User\_Field\_2]" \_

& " ,[Risk\_Owner\_M0],[Risk\_Owner\_Delegate\_M0]" \_

& " FROM [ERMDB].[dbo].[Risks\_Test]"

Call QueryERMDB(SQLstr, dataWS.Range(Col\_Letter(startCol) & 3), False)

End Sub

This grabs all of the columns out of the Controls Table. For more information please refer to the data dictionary.

Sub queryControls()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim queryPosition As Range

Dim indexRow As Long

Dim numCol As Long

Dim startCol As Long

Dim SQLstr As String

SQLstr = ""

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

numCol = dataIndexWS.Cells(indexRow, 3)

startCol = dataIndexWS.Cells(indexRow, 2)

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCol - 1) & 10000).Clear

SQLstr = SQLstr & "SELECT [Controls\_Primary\_key],[Risk\_ID],[Control\_ID]" \_

& ",[Control\_Description],[Control\_Owner],[Control\_Effectiveness]" \_

& ",[Control\_Effectiveness\_Custom],[Control\_Description\_Detail]" \_

& ",[Control\_Delegate],[User\_notes\_C1],[Sox],[Key\_Non\_Key]" \_

& ",[Custom\_Control\_ID],[Control\_Frequency],[Control\_Nature]" \_

& ",[Control\_Type],[Control\_Importance],[Monitoring\_Doc\_Description]" \_

& ",[Monitoring\_Description\_Location],[Monitoring\_Doc\_Signoff]" \_

& ",[Monitoring\_Doc\_Provider],[COSO\_Principle\_1],[COSO\_Principle\_2]" \_

& ",[COSO\_Principle\_3],[COSO\_Principle\_4],[COSO\_Principle\_5]" \_

& ",[COSO\_Principle\_6],[COSO\_Principle\_7],[COSO\_Principle\_8]" \_

& ",[COSO\_Principle\_9],[COSO\_Principle\_10],[COSO\_Principle\_11]"

SQLstr = SQLstr & ",[COSO\_Principle\_12],[COSO\_Principle\_13],[COSO\_Principle\_14]" \_

& ",[COSO\_Principle\_15],[COSO\_Principle\_16],[COSO\_Principle\_17]" \_

& ",[EY\_Reference],[Division],[Department],[Unit],[Process]" \_

& ",[SOC1],[Product],[System],[Regulation],[Control\_System]" \_

& ",[Financial\_Category],[Financial\_Line\_Item]" \_

& ",[SIW],[Audit\_Comments],[Monitoring\_Frequency]" \_

& ",[Date\_Last\_Monitored],[Date\_Last\_Attestation]" \_

& ",[Attestation\_Exceptions],[Shared\_Service\_Scenario]" \_

& ",[Shared\_Service\_Group],[Account],[Cost\_Center]" \_

& ",[User\_Field\_1],[User\_Field\_2],[Control\_Owner\_M0]" \_

& ",[Control\_Owner\_Delegate\_M0]" \_

& " FROM [ERMDB].[dbo].[Controls\_Test]"

Call QueryERMDB(SQLstr, dataWS.Range(Col\_Letter(startCol) & 3), False)

End Sub

This grabs the control delegates from the Controls Tables. This will be used to see how the email chain should work. Please refer to the data dictionary for more information.

Sub queryCOD()

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim queryPosition As Range

Dim indexRow As Long

Dim numCol As Long

Dim startCol As Long

Dim SQLstr As String

SQLstr = ""

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

numCol = dataIndexWS.Cells(indexRow, 3)

startCol = dataIndexWS.Cells(indexRow, 2)

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCol - 1) & 10000).Clear

SQLstr = "SELECT Distinct [Control\_Delegate] FROM [ERMDB].[dbo].[Controls\_Test]"

Call QueryERMDB(SQLstr, dataWS.Range(Col\_Letter(startCol) & 3), False)

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

dataWS.Range(Col\_Letter(startCol) & 3).Delete Shift:=xlUp

End If

End Sub

This moves objective ID and description over to the objectives worksheet from the data tab.

Sub objID\_Desc\_List()

Dim objWS As Worksheet

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set objWS = Sheets("Objectives")

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim startCol As Long

Dim numCols As Long

Dim objCount As Integer

Dim dataCount As Integer

Find starting columns and num of columns in data tab

startCol = dataIndexWS.Cells(WorksheetFunction.Match("Objective ID / Description", \_

dataIndexWS.Columns(1), 0), 2).Value

numCols = dataIndexWS.Cells(WorksheetFunction.Match("Objective ID / Description", \_

dataIndexWS.Columns(1), 0), 3).Value

Clear previous data

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCols - 1) & 10000).ClearContents

objCount = 0

dataCount = 3

Do While objWS.Cells(11 + objCount, 1) <> ""

dataWS.Cells(dataCount, startCol) = objWS.Cells(11 + objCount, 1)

dataWS.Cells(dataCount, startCol + 1) = objWS.Cells(11 + objCount, 2)

dataCount = dataCount + 1

objCount = objCount + 2

Loop

End Sub

This moves objective ID and description over to the objectives worksheet from the data tab.

Sub riskID\_Desc\_List()

Dim riskWS As Worksheet

Dim dataWS As Worksheet

Dim dataIndexWS As Worksheet

Set riskWS = Sheets("Risks")

Set dataWS = Sheets("Data Tab")

Set dataIndexWS = Sheets("Data Tab Index")

Dim startCol As Long

Dim numCols As Long

Dim riskCount As Integer

Dim dataCount As Integer

Find starting columns and number of columns in data tab

startCol = dataIndexWS.Cells(WorksheetFunction.Match("Risk ID / Description", \_

dataIndexWS.Columns(1), 0), 2).Value

numCols = dataIndexWS.Cells(WorksheetFunction.Match("Risk ID / Description", \_

dataIndexWS.Columns(1), 0), 3).Value

Clear previous data

dataWS.Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCols - 1) & 10000).ClearContents

riskCount = 0

dataCount = 3

Do While riskWS.Cells(11 + riskCount, 1) <> ""

dataWS.Cells(dataCount, startCol) = riskWS.Cells(11 + riskCount, 1)

dataWS.Cells(dataCount, startCol + 1) = riskWS.Cells(11 + riskCount, 2)

dataCount = dataCount + 1

riskCount = riskCount + 2

Loop

Range(Col\_Letter(startCol) & 3 & ":" & Col\_Letter(startCol + numCols - 1) & 10000).RemoveDuplicates Columns:=1, Header:=xlNo

End Sub