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

This is going to build strings to pull the data from the database to the worksheet using ADODB and SQL. We are currently still refining this to determine if we actually need the ADODB string or if the code below accomplishes what we need without users needing the ODBC connection on their machine.

sConnString = "Driver={SQL Server};Server={BTCSQLDBT350V};Database={ERMDB};" & \_

"Trusted\_Connection=Yes;"

This is kind of important, because if the users have to go and install odbc protocols, there will be a lot more questions and problems as we roll it out. This solution currently seems to allow users to use the worksheet without having to have an odbc connection.

Option Explicit

'Public Connection As ADODB.Connection

'Public Command As ADODB.Command

'Public RecordSet As ADODB.RecordSet

'Public Connecction As Object

'Public Command As Object

'Public RecordSet As Object

This is to see if we can ping the database.

Sub queryTest()

Dim SQLstr As String

SQLstr = "select \* from [ERMDB].[dbo].[Objectives\_Test]"

Call QueryERMDB(SQLstr, Sheets("Filter RCL").Range("B20"), False)

End Sub

Public Function QueryERMDB(strSQL As String, \_

position As Range, Optional Transpose As Boolean = False) As Boolean

Create the Connection and Recordset objects.

Dim conn As Object

Set conn = CreateObject("ADODB.Connection")

Dim rs As Object

Set rs = CreateObject("ADODB.RecordSet")

Dim sConnString As String

Dim result As Variant

On Error GoTo QueryERMDB\_Err

Create the connection string.

'sConnString = strConnect ' Usually "ERMDB" <- This is a backup system. Line below is what is used.

sConnString = "Driver={SQL Server};Server={BTCSQLDBT350V};Database={ERMDB};" & \_

"Trusted\_Connection=Yes;"

Open the connection and execute.

conn.Open sConnString

Const adStateOpen As Long = 1

Set rs = conn.Execute(strSQL) 'Test string: "SELECT \* FROM [RiskManagement].[dbo].[MI\_Main];"

Check we have data.

If Not rs.EOF Then

' Transfer result.

If Transpose = True Then

result = rs.GetRows

position.Resize(UBound(result, 1) + 1, UBound(result, 2) + 1).Value = result

Else

position.CopyFromRecordset rs

End If

Close the recordset

rs.Close

Else

'MsgBox "Error: No records returned.", vbCritical

End If

Clean up

If CBool(conn.State And adStateOpen) Then conn.Close

Set conn = Nothing

Set rs = Nothing

QueryERMDB = True

QueryERMDB\_End:

On Error Resume Next

rs.Close

Set rs = Nothing

Exit Function

In case of error

QueryERMDB\_Err:

QueryERMDB = False

MsgBox "Error: " & err.Number & vbCrLf & err.Description

Resume QueryERMDB\_End:

End Function

If it works, then it works. If the error code kicks, then there’s a problem.

Let’s connect to the server. This only opens the connection, it doesn’t actually pull anything out.

Public Function UploadERMDB(Query As String)

Dim strCommand As String

Dim conn As Object

Dim sConnString As String

Dim Command As Object

Set Command = CreateObject("ADODB.Command")

sConnString = "Driver={SQL Server};Server={BTCSQLDBT350V};Database={ERMDB};" & \_

"Trusted\_Connection=Yes;"

Set conn = CreateObject("ADODB.Connection")

Open the connection

conn.Open sConnString

strCommand = Query

With Command

.ActiveConnection = conn

.CommandTimeout = 1000000000

.CommandText = strCommand

.Execute

End With

End Function

This is all old code that we aren’t sure if we need to use later or not. Still developing.

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

'DataBaseConnection doesn't appear to be needed

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

'Public Function DataBaseConnection()

'Set Connection = New ADODB.Connection

'Set RecordSet = New ADODB.RecordSet

'Set Command = New ADODB.Command

'

'On Error GoTo ErrorDataBaseConnection

'

'With Connection

' .Provider = "MSDASQL"

' .ConnectionString = "ERMDB"

' .ConnectionTimeout = 1000000000

' .Open

'End With

'

'ErrorDataBaseConnection:

' If err Then MsgBox "Please contact the system administrator. ", vbOKOnly, "Function DataBaseConnection"

'End Function