**File: Auto\_Query\_Host\_r7E**

**Module 8**

Function Highlight\_Finder\_Match(ByVal currentRow As Integer)

'function takes the currently selected highlight color and uses that as its highlighting color

'declare vars

Dim highlightColor As String

'initialize vars

With Worksheets("AutoQuery").Range("H2")

highlightColor = CStr(Trim(.Value))

End With

'change color of selection

With Worksheets("AutoQuery").Range("B" & CStr(currentRow) & ":M" & CStr(currentRow))

If (StrComp("Yellow", highlightColor) = 0) Then

.Interior.Color = vbYellow

ElseIf (StrComp("Red", highlightColor) = 0) Then

.Interior.Color = vbRed

ElseIf (StrComp("Magenta", highlightColor) = 0) Then

.Interior.Color = vbMagenta

ElseIf (StrComp("Green", highlightColor) = 0) Then

.Interior.Color = vbGreen

ElseIf (StrComp("Cyan", highlightColor) = 0) Then

.Interior.Color = vbCyan

ElseIf (StrComp("Blue", highlightColor) = 0) Then

.Interior.Color = vbBlue

End If

End With

End Function 'end of highlight finder match

Sub Finder\_Button()

'Finder button searches for members with matching product codes (if they exist on host)

'Prompts user for product code , then searches for them

'Highlights any matches using the selected highlighter color

'disable features to optimize macro for speed

Call disableFeaturesForMacroOptimization

'check to see if host is set-------------------------------------------

If (SetHost = False) Then

MsgBox "ERROR: Host not Set. Make sure Host is open."

Exit Sub

End If

'----------------------------------------------------------------------

'declare variables

Dim seekingProductCodeValue As String 'holds product code value to look for

Dim finished As Boolean

Dim index As Integer

Dim lastPageOfAccounts As Boolean

Dim accountType As String

Dim ProductCode As String

Dim accountStatus As String

Dim valueToQuery As String

Dim valueType As String

Dim counter As Integer

Dim firstAccountOnPrevPage As String 'used to determine when you are on the last page

Dim checkForActiveStatus As Boolean

'initialize variables

seekingProductCodeValue = ""

finished = False

index = 5 'start at row 5

lastPageOfAccounts = False

accountType = ""

ProductCode = ""

accountStatus = ""

valueToQuery = ""

valueType = ""

counter = 0

firstAccountOnPrevPage = ""

checkForActiveStatus = False

'inputbox prompt for product code-------------

seekingProductCodeValue = InputBox("Enter in the product code you are looking for:")

seekingProductCodeValue = Trim(seekingProductCodeValue) 'trim input

seekingProductCodeValue = UCase(seekingProductCodeValue) 'convert to uppercase if alpha value

'MsgBox "product code:" & CStr(seekingProductCodeValue)

'end of inputbox for product code-------------

'prompt to only display accounts with active status----------

Dim activeStatusPrompt As Integer

activeStatusPrompt = MsgBox("Do you want only the matching accounts that have ACTIVE status to be displayed?", vbYesNo + vbQuestion, "Display only ACTIVE status matches?")

If activeStatusPrompt = vbYes Then

checkForActiveStatus = True

Else

checkForActiveStatus = False

End If

'end of prompt to onl;y display accounts with active status------

'Display the Finder Form----------------------------

'Finder\_Form.Show (seekingProductCodeValue)

'End of display the finder form---------------------

'make sure inputbox isnt empty, or user hit cancel

If (StrComp(vbNullString, seekingProductCodeValue) = 0) Then

Exit Sub

End If

'end of make sure inputbox isnt empty, or user hit cancel

'make sure there are values to check---------------

With Worksheets("AutoQuery").Range("A5")

If (StrComp(vbNullString, Trim(.Value)) = 0) Then

MsgBox "Error: You must have at least one member to check"

Exit Sub

End If

End With

'end of make sure there are values to check--------

'load valueType

With Worksheets("AutoQuery").Range("A2")

valueType = Trim(.Value)

End With

'while not at end of list, check to see if member has matching product code-------------

Do While (finished = False)

'check to see if cell is blank

With Worksheets("AutoQuery").Range("A" & CStr(index))

If (StrComp(vbNullString, Trim(.Value)) = 0) Then 'exit clause

finished = True

Exit Do

Else

'load value to query

valueToQuery = Trim(.Value)

End If

End With

'check to make sure member exists

Dim exists As String

With Worksheets("AutoQuery").Range("B" & CStr(index))

exists = Trim(.Value)

End With

'check to see if exists field is blank

If (StrComp(vbNullString, exists) = 0) Then

'break from loop

MsgBox "Error: Please Run Auto Query Host Before Running Finder"

finished = True

Exit Do

End If 'end of if exists field is blank

If (StrComp("True", exists) = 0) Then

'If there is a SSN/CIS conflict, skip checking that member

Dim conflictCheck As String

Dim conflictCheckSum As Integer

With Worksheets("AutoQuery").Range("C" & CStr(index))

conflictCheck = Trim(.Value)

End With

conflictCheckSum = InStr(1, conflictCheck, "Error: Multiple Members have", vbTextCompare) 'returns 0 if substring is not found in the string

If (conflictCheckSum = 0) Then 'if there is not an ssn cis error

Call ResetForCommand

If (StrComp(valueType, "CIS") = 0) Then 'if CIS

HostSendCommand ("rmab;nb" & valueToQuery)

ElseIf (StrComp(valueType, "SSN/TIN") = 0) Then 'if SSN/TIN

HostSendCommand ("rmab;tn" & valueToQuery)

Else

'throw error

MsgBox "Error: Only SSN/TIN and CIS valueTypes are supported for Finder"

finished = True

Exit Do

End If 'end of if inputValueType is CIS

'Product Code - fields 52, 75, 98, 121, 144, 167, 190

'Account Status - fields 59, 82, 105, 128, 151, 174, 197

'check all of members accounts-----------------------------

Do While (lastPageOfAccounts = False)

'part 1-----------------

'check to see if first account is same as last pages first account

Dim lastPageCheckString As String

lastPageCheckString = Trim(CurrentHost.Fields(45))

If (StrComp(firstAccountOnPrevPage, lastPageCheckString) = 0) Then

'break out of loop

lastPageOfAccounts = True

Exit Do

End If 'end of if lastPageErrorValue like ....

'end of part 1------------------

'part 2-----------------------

'Check if Account 1 is blank, if not then extract info

accountType = Trim(CurrentHost.Fields(44))

If Not (StrComp(accountType, vbNullString) = 0) Then 'if not blank, get data

ProductCode = Trim(CurrentHost.Fields(52))

accountStatus = Trim(CurrentHost.Fields(59))

'set firstAccountOnPrevPage value

firstAccountOnPrevPage = Trim(CurrentHost.Fields(45)) 'account number field

'check to see if matches finder

If (StrComp(ProductCode, seekingProductCodeValue) = 0) Then

If (checkForActiveStatus = True) Then

If (StrComp(accountStatus, "ACTIVE") = 0) Then

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'account status is active

Else

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'end of if checkForActiveStatus = true

'MsgBox "A match was found for member"

lastPageOfAccounts = True

Exit Do

End If 'end of if product matches seekingproductcodevalue

End If 'end of if account 1 is not blank

'check if account 2 is not blank

accountType = Trim(CurrentHost.Fields(67))

If Not (StrComp(accountType, vbNullString) = 0) Then 'if not blank, get data

ProductCode = Trim(CurrentHost.Fields(75))

accountStatus = Trim(CurrentHost.Fields(82))

'check to see if matches finder

If (StrComp(ProductCode, seekingProductCodeValue) = 0) Then

If (checkForActiveStatus = True) Then

If (StrComp(accountStatus, "ACTIVE") = 0) Then

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'account status is active

Else

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'end of if checkForActiveStatus = true

'MsgBox "A match was found for member"

lastPageOfAccounts = True

Exit Do

End If 'end of if product matches seekingproductcodevalue

End If 'end of if account 2 is not blank

'check if account 3 is not blank

accountType = Trim(CurrentHost.Fields(90))

If Not (StrComp(accountType, vbNullString) = 0) Then 'if not blank, get data

ProductCode = Trim(CurrentHost.Fields(98))

accountStatus = Trim(CurrentHost.Fields(105))

'check to see if matches finder

If (StrComp(ProductCode, seekingProductCodeValue) = 0) Then

If (checkForActiveStatus = True) Then

If (StrComp(accountStatus, "ACTIVE") = 0) Then

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'account status is active

Else

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'end of if checkForActiveStatus = true

'MsgBox "A match was found for member"

lastPageOfAccounts = True

Exit Do

End If 'end of if product matches seekingproductcodevalue

End If 'end of if account 3 is not blank

'check if account 4 is not blank

accountType = Trim(CurrentHost.Fields(113))

If Not (StrComp(accountType, vbNullString) = 0) Then 'if not blank, get data

ProductCode = Trim(CurrentHost.Fields(121))

accountStatus = Trim(CurrentHost.Fields(128))

'check to see if matches finder

If (StrComp(ProductCode, seekingProductCodeValue) = 0) Then

If (checkForActiveStatus = True) Then

If (StrComp(accountStatus, "ACTIVE") = 0) Then

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'account status is active

Else

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'end of if checkForActiveStatus = true

'MsgBox "A match was found for member"

lastPageOfAccounts = True

Exit Do

End If 'end of if product matches seekingproductcodevalue

End If 'end of if account 4 is not blank

'check if account 5 is not blank

accountType = Trim(CurrentHost.Fields(136))

If Not (StrComp(accountType, vbNullString) = 0) Then 'if not blank, get data

ProductCode = Trim(CurrentHost.Fields(144))

accountStatus = Trim(CurrentHost.Fields(151))

'check to see if matches finder

If (StrComp(ProductCode, seekingProductCodeValue) = 0) Then

If (checkForActiveStatus = True) Then

If (StrComp(accountStatus, "ACTIVE") = 0) Then

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'account status is active

Else

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'end of if checkForActiveStatus = true

'MsgBox "A match was found for member"

lastPageOfAccounts = True

Exit Do

End If 'end of if product matches seekingproductcodevalue

End If 'end of if account 5 is not blank

'check if account 6 is not blank

accountType = Trim(CurrentHost.Fields(159))

If Not (StrComp(accountType, vbNullString) = 0) Then 'if not blank, get data

ProductCode = Trim(CurrentHost.Fields(167))

accountStatus = Trim(CurrentHost.Fields(174))

'check to see if matches finder

If (StrComp(ProductCode, seekingProductCodeValue) = 0) Then

If (checkForActiveStatus = True) Then

If (StrComp(accountStatus, "ACTIVE") = 0) Then

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'account status is active

Else

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'end of if checkForActiveStatus = true

'MsgBox "A match was found for member"

lastPageOfAccounts = True

Exit Do

End If 'end of if product matches seekingproductcodevalue

End If 'end of if account 6 is not blank

'check if account 7 is not blank

accountType = Trim(CurrentHost.Fields(182))

If Not (StrComp(accountType, vbNullString) = 0) Then 'if not blank, get data

ProductCode = Trim(CurrentHost.Fields(190))

accountStatus = Trim(CurrentHost.Fields(197))

'check to see if matches finder

If (StrComp(ProductCode, seekingProductCodeValue) = 0) Then

If (checkForActiveStatus = True) Then

If (StrComp(accountStatus, "ACTIVE") = 0) Then

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'account status is active

Else

Call Highlight\_Finder\_Match(index)

counter = counter + 1

End If 'end of if checkForActiveStatus = true

'MsgBox "A match was found for member"

lastPageOfAccounts = True

Exit Do

End If 'end of if product matches seekingproductcodevalue

End If 'end of if account 7 is not blank

'end of part 2-----------------

HostSendCommand ("@1") 'press F1 to goto the next page

Loop 'end of while lastPageOfAccounts is False loop

End If 'end of if member exists

'end of check all of members accounts-----------------------

End If 'End of if there is a SSN/CIS conflict, skip checking member

'reset lastPageOfAccounts Value

lastPageOfAccounts = False

'increment index

index = index + 1

Loop 'end of loop for finished checking all members

'end of while loop-----------------------

'display finished message

MsgBox "Finished searching for product code: " & CStr(seekingProductCodeValue) & ". Found " & CStr(counter) & " members with at least one matching account."

're-enable features once macro is finished

Call enableFeaturesAfterOptimizedMacroIsDone

End Sub 'end of finder button