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

**Module 1**

'===============================================================

'COPIED FROM HOST\_SCRAPER IN SEARCH TOOL

'DECLARATION OF CUSTOM TYPES

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

Option Explicit

Private HE As HostExApplication

Public CurrentHost As HostExHost

'============================================================

Public Function SetHost(Optional ByRef NewHost As HostExHost)

'Overview:

'Sets the last HostExplorer(TSSO) that was in focus to the HostExplorer object that the HostExplorer API will be using

'

'Parameters(Optional):

'A specific HostExplorer instance given by name

'

'Pre-Condition:

'A HostExplorer(TSSO) instance is open

'

'Post-Condition:

'A HostExplorer (TSSO) instance will be put into focus for the HostExplorer API

'

Set HE = New HostExApplication

If NewHost Is Nothing Then

Set CurrentHost = HE.CurrentHost

Else

Set CurrentHost = NewHost

End If

If CurrentHost Is Nothing Then

SetHost = False

Else

SetHost = True

End If

End Function 'end of SetHost

Public Function LoggedIntoHost() As Boolean

'Overview:

'Determines if a user is currently logged into Host(TSSO)

'

'Parameters:

'None

'

'Precondition:

'SetHost has been called and has returned True

'

'Post-Condition:

'If the user is logged into a HostExplorer (TSSO) instance Return True, otherwise Return False

'

Call ResetForCommand

HostSendCommand ("RMLP")

If CurrentHost.Fields(1) = "RMLP" Then

LoggedIntoHost = True

Else

LoggedIntoHost = False

End If

End Function 'end of LoggedIntoHost

Public Function ResetForCommand()

'Overview:

'Sends key strokes to the HostExplorer (TSSO) instance that clears the screen and puts the cursor at the top left of the screen

'(simulates a user clicking the ESC key)

'

'Parameters:

'None

'

'Pre-Condition:

'A HostExplorer (TSSO) instance is open and in focus

'

'Post-Condition:

'The screen clears and is ready for an input (command)

'

'Sends clear twice (or ESC),

'Then clicks Home - should reset from any screen

HostSendKeys ("@C@C@0")

End Function

Public Function HostSendCommand(ByVal Keys As String)

'Overview:

'Enters the given keys into a HostExplorer (TSSO) instance and clicks Enter

'(Submits the given keys)

'

'Parameters:

'A string of valid key entries. Example: ("@0RMAB") -> Clicks Home, Enters "RMAB"

'

'Pre-Condition:

'A HostExplorer (TSSO) instance is open and in focus

'

'Post-Condition:

'The given keys are entered into the HostExplorer (TSSO) instance and the Enter key is clicked

'

CurrentHost.Keys (Keys & "@E")

End Function

Public Function HostSendKeys(ByVal Keys As String)

'Overview:

'Enters the given keys into a HostExplorer (TSSO) instance

'

'Parameters:

'A string of key entries. Example: ("@E") -> Clicks Enter

'

'Pre-Condition:

'A HostExplorer (TSSO) instance is open and in focus

'

'Post-Condition:

'The given keys are entered into the HostExplorer (TSSO) instance

'

CurrentHost.Keys (Keys)

End Function

Function disableFeaturesForMacroOptimization()

'disable features that are not needed, for optimization

With Application

.EnableEvents = False: .Calculation = xlCalculationManual: .ScreenUpdating = False

.DisplayAlerts = False: .DisplayStatusBar = False

End With

With ActiveSheet

.DisplayPageBreaks = False

End With

End Function 'end of disableFeaturesForMacroOptimization

Function enableFeaturesAfterOptimizedMacroIsDone()

're-enable features that where disabled for speed

With Application

.EnableEvents = True: .ScreenUpdating = True: .DisplayAlerts = True: .DisplayStatusBar = True

End With

End Function 'end of enableFeaturesAfterOptimzed macro is done

Public Function retrieveMemberRMI1Info(ByVal valueToQuery As String, ByVal valueType As String, ByRef memberName As String, ByRef memberSSN As String, ByRef memberCIS As String, ByRef memberBirthday, ByRef memberStreetAddress As String, ByRef memberCityStateAddress As String, ByRef memberZipAddress As String) As Boolean

'Function returns false if member does not exist on host

Dim strCurrentScreen As String

Call SetHost

'defualt to True

retrieveMemberRMI1Info = True

'trim valueToQuery and valueType

valueToQuery = Trim(valueToQuery)

valueType = Trim(valueType)

Call ResetForCommand

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

HostSendCommand ("rmi1;nb" & valueToQuery)

Else 'if SSN/TIN

HostSendCommand ("rmi1;tn" & valueToQuery)

End If 'end of if inputValueType is CIS

'determine if member exists

strCurrentScreen = CurrentHost.Fields(1)

If strCurrentScreen = "RMLP" Then

retrieveMemberRMI1Info = False

Exit Function

ElseIf strCurrentScreen = "RMLL" Then 'if there are multiple SSN/TINs for this value being queried

retrieveMemberRMI1Info = True

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

memberName = "Error: Multiple Members have this CIS. Try querying with SSN/TIN instead or manually searching Host."

Else 'if SSN/TIN

memberName = "Error: Multiple Members have this SSN/TIN. Try querying with CIS instead or manually searching Host."

End If 'end of if CIS

memberSSN = ""

memberCIS = ""

memberBirthday = ""

memberStreetAddress = ""

memberCityStateAddress = ""

memberZipAddress = ""

Exit Function

End If 'end of if determine if member exists (or if there is an SSN/TIN conflict)

'retrieve info from page RMI1------------------------------------

' Name - Field 8

' SSN/TIN - Field 9

' CIS - Field 13

' BirthDate - Field 36

' Street Address - Field 25

' City/State Address - Field 46

' Zip Address - Field 49

memberName = Trim(CurrentHost.Fields(8)) 'Name

memberSSN = Trim(CurrentHost.Fields(9)) 'SSN/TIN

memberCIS = Trim(CurrentHost.Fields(13)) 'CIS

memberBirthday = Trim(CurrentHost.Fields(36)) 'Birthdate

memberStreetAddress = Trim(CurrentHost.Fields(25)) 'Street Address

memberCityStateAddress = Trim(CurrentHost.Fields(46)) 'City/State

memberZipAddress = Trim(CurrentHost.Fields(49)) 'Zip

'END of retrieve info from RMI1---------------------------

'Dim i As Integer 'display the fields

'For i = 1 To CurrentHost.Fields.Count - 1

'column g is index, column h is value

' With Worksheets("AutoQuery").Range("G" & CStr(i))

' .Value = CStr(i)

' End With

' With Worksheets("AutoQuery").Range("H" & CStr(i))

' .Value = CStr(CurrentHost.Fields(i))

' End With

'Next i 'end of for loop

End Function 'end of retrieveMemberRMI1Info

Function displayMemberRMI1InfoHeaders()

With Worksheets("AutoQuery").Range("A4")

.Value = "Value's to Query": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("B4")

.Value = "Exists on Host": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("C4")

.Value = "Name": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("D4")

.Value = "Street Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("E4")

.Value = "City/State Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("F4")

.Value = "Zip": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("G4")

.Value = "Est/Birth Day": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("H4")

.Value = "CIS": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("I4")

.Value = "SSN/TIN": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

'clear unused header cells

With Worksheets("AutoQuery").Range("J4:AA4")

.Value = ""

End With

End Function 'end of displayMemberRMI1InfoHeaders

Function displayMemberRMI1InfoResults(ByVal index, ByVal existingValue As Boolean, ByVal memberName As String, ByVal memberStreetAddress As String, ByVal memberCityStateAddress As String, ByVal memberZipAddress As String, ByVal memberBirthday As String, ByVal memberCIS As String, ByVal memberSSN As String)

If (existingValue = True) Then

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

.Value = CStr(existingValue)

End With

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

.Value = memberName

End With

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

.Value = memberStreetAddress

End With

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

.Value = memberCityStateAddress

End With

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

.Value = memberZipAddress

End With

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

.Value = memberBirthday

End With

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

.Value = memberCIS

End With

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

.Value = memberSSN

End With

Else 'if false

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

.Value = CStr(existingValue)

End With

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

.Value = "N/A"

End With

End If

End Function 'end of displayMemberRMI1InfoHeaders

Public Function retrieveMemberIMI1Info(ByVal valueToQuery As String, ByVal valueType As String, ByRef accountProductCode As String, ByRef accountType As String, ByRef accountStatus As String, ByRef accountPorB As String, ByRef accountBalance As String, ByRef accountLoanBalance As String, ByRef memberName As String, ByRef memberStreetAddress As String, ByRef memberCityStateAddress As String, ByRef memberSSN As String) As Boolean

Call SetHost

'defualt to True

retrieveMemberIMI1Info = True

'trim valueToQuery and valueType

valueToQuery = Trim(valueToQuery)

valueType = Trim(valueType)

Call ResetForCommand

HostSendCommand ("imi1;" & valueToQuery)

'determine if member exists and get values

'Product Code - Field 26

'Account Type - Field 27

'Account Status - Field 30

'Personal or Commrcial Account - Field 32

'Account Owners Name - Field 33 (Member name)

'Account Owners Street Address (or 2nd account owner) - Field 36

'Account Balance - Field 38

'Account Owners street Address (or 3rd account owner) - field 39

'Account Owners street address (or 4th account owner) - field 42

'account owners street address (if 4 account owners) - field 45

'account owners city/state address (if 4 accounts owners) - field 48

'Account Loan Balance - Field 47

'Account Owners SSN - Field 93

accountProductCode = Trim(CurrentHost.Fields(26))

accountType = Trim(CurrentHost.Fields(27))

accountStatus = Trim(CurrentHost.Fields(30))

accountPorB = Trim(CurrentHost.Fields(32))

accountBalance = Trim(CurrentHost.Fields(38))

accountLoanBalance = Trim(CurrentHost.Fields(47))

memberSSN = Trim(CurrentHost.Fields(93))

'deterime number of account owners---------------------------------------------

Dim numOfAccountOwners As Integer

Dim firstEntry As String

Dim secondEntry As String

Dim thirdEntry As String

Dim fourthEntry As String

Dim fifthEntry As String

Dim sixthEntry As String

firstEntry = Trim(CurrentHost.Fields(33))

secondEntry = Trim(CurrentHost.Fields(36))

thirdEntry = Trim(CurrentHost.Fields(39))

fourthEntry = Trim(CurrentHost.Fields(42))

fifthEntry = Trim(CurrentHost.Fields(45))

sixthEntry = Trim(CurrentHost.Fields(48))

numOfAccountOwners = getNumberOfAccountOwners(firstEntry, secondEntry, thirdEntry, fourthEntry, fifthEntry, sixthEntry)

'end of determin number of account owners--------------------------------------------

'if 1 account owner

If (numOfAccountOwners = 1) Then

memberName = firstEntry

memberStreetAddress = secondEntry

memberCityStateAddress = thirdEntry

ElseIf (numOfAccountOwners = 2) Then

memberName = firstEntry & ", " & secondEntry

memberStreetAddress = thirdEntry

memberCityStateAddress = fourthEntry

ElseIf (numOfAccountOwners = 3) Then

memberName = firstEntry & ", " & secondEntry & ", " & thirdEntry

memberStreetAddress = fourthEntry

memberCityStateAddress = fifthEntry

Else 'if 4

memberName = firstEntry & ", " & secondEntry & ", " & thirdEntry & ", " & fourthEntry

memberStreetAddress = fifthEntry

memberCityStateAddress = sixthEntry

End If

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

retrieveMemberIMI1Info = False

End If

'Dim i As Integer 'display the fields

'For i = 6 To CurrentHost.Fields.Count - 1

' 'column g is index, column h is value

' With Worksheets("AutoQuery").Range("G" & CStr(i))

' .Value = CStr(i)

' End With

' With Worksheets("AutoQuery").Range("H" & CStr(i))

' .Value = CStr(CurrentHost.Fields(i))

' End With

'Next i 'end of for loop

End Function 'end of retrieveMemberIMI1Info

Function displayMemberIMI1InfoHeaders()

With Worksheets("AutoQuery").Range("A4")

.Value = "Value's to Query": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("B4")

.Value = "Exists on Host": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("C4")

.Value = "Account Owner's Name": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("D4")

.Value = "Account Owner's Street Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("E4")

.Value = "Account Owner's City/State/Zip Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("F4")

.Value = "SSN/TIN": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("G4")

.Value = "Account Product Code": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("H4")

.Value = "Account Type": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("I4")

.Value = "Account Status": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("J4")

.Value = "Personal or Commercial Account": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("K4")

.Value = "Account Balance": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("L4")

.Value = "Account Loan Balance": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

'clear unused header cells

With Worksheets("AutoQuery").Range("M4:AA4")

.Value = ""

End With

End Function 'end of displayMemberIMI1InfoHeaders

Function displayMemberIMI1InfoResults(ByVal index As Integer, ByVal existingValue As Boolean, ByVal memberName As String, ByVal memberStreetAddress As String, ByVal memberCityStateAddress As String, ByVal memberSSN As String, ByVal accountProductCode As String, ByVal accountType As String, ByVal accountStatus As String, ByVal accountPorB As String, ByVal accountBalance As String, ByVal accountLoanBalance As String)

If (existingValue = True) Then

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

.Value = CStr(existingValue)

End With

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

.Value = memberName

End With

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

.Value = memberStreetAddress

End With

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

.Value = memberCityStateAddress

End With

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

.Value = memberSSN

End With

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

.Value = accountProductCode

End With

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

.Value = accountType

End With

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

.Value = accountStatus

End With

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

.Value = accountPorB

End With

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

.Value = accountBalance

End With

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

.Value = accountLoanBalance

End With

Else 'if false

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

.Value = CStr(existingValue)

End With

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

.Value = "N/A"

End With

End If

End Function 'end of dislpayMemberIMI1InfoResults

Function displayMemberSTI1InfoHeaders()

With Worksheets("AutoQuery").Range("A4")

.Value = "Value's to Query": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("B4")

.Value = "Exists on Host": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("C4")

.Value = "Account Owner's Name(s)": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("D4")

.Value = "Account Owner's Street Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("E4")

.Value = "Account Owner's City/State/Zip Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("F4")

.Value = "SSN/TIN": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("G4")

.Value = "CIS": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("H4")

.Value = "Account Product Code": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("I4")

.Value = "Account Type": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("J4")

.Value = "Account Status": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("K4")

.Value = "Personal or Commercial Account": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("L4")

.Value = "Account Memo Balance": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("M4")

.Value = "Account Current Balance": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("N4")

.Value = "Account Holds": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

'clera the rest of the unsed headers

With Worksheets("AutoQuery").Range("O4:AA4")

.Value = ""

End With

End Function 'end of displayMembersSTI1InfoHeaders

Public Function retrieveMemberSTI1Info(ByRef valueToQuery As String, ByRef valueType As String, ByRef accountProductCode As String, ByRef accountType As String, ByRef memberName As String, ByRef accountMemoBalance As String, ByRef memberStreetAddress As String, ByRef accountCurrentBalance As String, ByRef memberCityStateAddress As String, ByRef accountHolds As String, ByRef memberCIS As String, ByRef memberSSN As String, ByRef accountStatus As String, ByRef accountPorB As String) As Boolean

Call SetHost

'defualt to True

retrieveMemberSTI1Info = True

'trim valueToQuery and valueType

valueToQuery = Trim(valueToQuery)

valueType = Trim(valueType)

Call ResetForCommand

HostSendCommand ("sti1;" & valueToQuery)

'determine if member exists and get values

'Product Code - Field 23

'Account Type - Field 24

'Account Owner Name - Field 31

'Account Memo Balance - field 33

'Account Owner Street Address (or 2nd account owner) - Field 34

'Account current balance - field 36

'Account Owner City State Address (or 3rd account owner) - Field 37

'account owner city state address (if 2 account owners) - field 40

'account owner city state address (if 3 account owners) - field 43

'account owner city state address (if 4 account owners) - field 46

'account holds - field 45

'Account Owner CIS - Field 50

'Account Owner SSN - Field 55

'Account Status - field 75

'a personal or business account - field 101

accountProductCode = Trim(CurrentHost.Fields(23))

accountType = Trim(CurrentHost.Fields(24))

accountMemoBalance = Trim(CurrentHost.Fields(33))

accountCurrentBalance = Trim(CurrentHost.Fields(36))

accountHolds = Trim(CurrentHost.Fields(45))

memberCIS = Trim(CurrentHost.Fields(50))

memberSSN = Trim(CurrentHost.Fields(55))

accountStatus = Trim(CurrentHost.Fields(75))

accountPorB = Trim(CurrentHost.Fields(101))

'deterime number of account owners---------------------------------------------------------------

Dim numOfAccountOwners As Integer

Dim firstEntry As String

Dim secondEntry As String

Dim thirdEntry As String

Dim fourthEntry As String

Dim fifthEntry As String

Dim sixthEntry As String

firstEntry = Trim(CurrentHost.Fields(31))

secondEntry = Trim(CurrentHost.Fields(34))

thirdEntry = Trim(CurrentHost.Fields(37))

fourthEntry = Trim(CurrentHost.Fields(40))

fifthEntry = Trim(CurrentHost.Fields(43))

sixthEntry = Trim(CurrentHost.Fields(46))

numOfAccountOwners = getNumberOfAccountOwners(firstEntry, secondEntry, thirdEntry, fourthEntry, fifthEntry, sixthEntry)

'end of determin number of account owners---------------------------------------------------

'if 1 account owner

If (numOfAccountOwners = 1) Then

memberName = firstEntry

memberStreetAddress = secondEntry

memberCityStateAddress = thirdEntry

ElseIf (numOfAccountOwners = 2) Then

memberName = firstEntry & ", " & secondEntry

memberStreetAddress = thirdEntry

memberCityStateAddress = fourthEntry

ElseIf (numOfAccountOwners = 3) Then

memberName = firstEntry & ", " & secondEntry & ", " & thirdEntry

memberStreetAddress = fourthEntry

memberCityStateAddress = fifthEntry

Else 'if 4

memberName = firstEntry & ", " & secondEntry & ", " & thirdEntry & ", " & fourthEntry

memberStreetAddress = fifthEntry

memberCityStateAddress = sixthEntry

End If

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

retrieveMemberSTI1Info = False

End If

'Dim i As Integer 'display the fields

'For i = 6 To CurrentHost.Fields.Count - 1

'column g is index, column h is value

' With Worksheets("AutoQuery").Range("G" & CStr(i))

' .Value = CStr(i)

' End With

' With Worksheets("AutoQuery").Range("H" & CStr(i))

' .Value = CStr(CurrentHost.Fields(i))

' End With

'Next i 'end of for loop

End Function 'end of retrieveMemberSTI1Info

Function displayMemberSTI1InfoResults(ByVal index As Integer, ByVal existingValue As Boolean, ByVal memberName As String, ByVal memberStreetAddress As String, ByVal memberCityStateAddress As String, ByVal memberSSN As String, ByVal memberCIS As String, ByVal accountProductCode As String, ByVal accountType As String, ByVal accountStatus As String, ByVal accountPorB As String, ByVal accountMemoBalance As String, ByVal accountCurrentBalance As String, ByVal accountHolds As String)

If (existingValue = True) Then

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

.Value = CStr(existingValue)

End With

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

.Value = memberName

End With

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

.Value = memberStreetAddress

End With

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

.Value = memberCityStateAddress

End With

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

.Value = memberSSN

End With

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

.Value = memberCIS

End With

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

.Value = accountProductCode

End With

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

.Value = accountType

End With

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

.Value = accountStatus

End With

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

.Value = accountPorB

End With

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

.Value = accountMemoBalance

End With

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

.Value = accountCurrentBalance

End With

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

.Value = accountHolds

End With

Else 'if doesnt exist

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

.Value = CStr(existingValue)

End With

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

.Value = "N/A"

End With

End If

End Function 'end of displayMemberSTI1InfoResults

Function displayMemberAMAIInfoHeaders()

With Worksheets("AutoQuery").Range("A4")

.Value = "Value's to Query": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("B4")

.Value = "Exists on Host": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("C4")

.Value = "Account Owner's Name": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("D4")

.Value = "Account Owner's Street Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("E4")

.Value = "Account Owner's City/State/Zip Address": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("F4")

.Value = "Account Owner's SSN/TIN": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("G4")

.Value = "Account Product Code": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("H4")

.Value = "Account Status": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("I4")

.Value = "Payoff Amount": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("J4")

.Value = "Original Loan Amount": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("K4")

.Value = "Current Rate": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("L4")

.Value = "Original Proceeds": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("M4")

.Value = "Current Principle": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("N4")

.Value = "Contract Date": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("O4")

.Value = "Current Maturity Date": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("P4")

.Value = "Current Term": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("Q4")

.Value = "Scheduled Payment Amount": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("R4")

.Value = "Closed Date": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("S4")

.Value = "Payments Made": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("T4")

.Value = "Current Payment Amount": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("U4")

.Value = "Scheduled Due Date": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("V4")

.Value = "Past Due Amount": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("W4")

.Value = "Collateral Code": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

With Worksheets("AutoQuery").Range("X4")

.Value = "Desciption": .Font.Bold = True: .Font.Underline = True: .Font.Color = vbWhite: .Interior.Color = vbBlack

End With

'clear unused headers

With Worksheets("AutoQuery").Range("Y4:AA4")

.Value = ""

End With

End Function 'end of displayMemberAMAIInfoHeaders

Public Function retrieveMemberAMAIInfo(ByRef valueToQuery As String, ByRef valueType As String, ByRef memberName As String, ByRef memberStreetAddress As String, ByRef memberCityStateAddress As String, ByRef memberSSN As String, ByRef accountProductCode As String, ByRef accountStatus As String, ByRef accountPayoffAmount As String, ByRef accountOriginalLoanAmount As String, ByRef accountCurrentRate As String, ByRef accountOriginalProceeds As String, ByRef accountCurrentPrinciple As String, ByRef accountContractDate As String, ByRef accountCurrentMaturityDate As String, ByRef accountCurrentTerm As String, ByRef accountScheduledPaymentAmount As String, ByRef accountClosedDate As String, ByRef accountPaymentsMade As String, ByRef accountCurrentPaymentAmount As String, ByRef accountScheduledDueDate As String, ByRef accountPastDueAmount As String, ByRef accountCollateralCode As String, ByRef accountDescription As String) As Boolean

Call SetHost

'defualt to True

retrieveMemberAMAIInfo = True

'trim valueToQuery and valueType

valueToQuery = Trim(valueToQuery)

valueType = Trim(valueType)

Call ResetForCommand

HostSendCommand ("amai") 'must goto amai screen first, then load the account

HostSendCommand (valueToQuery)

'determine if member exists and get values

''Account Owner SSN - Field 27

''Account Status - Field 29

''Payoff Amount - Field 33

''Original Loan Amount - Field 40

''Current Rate - Field 42

''Account Product Code - Field 44

''Original Proceeds - Field 46

''Current Principle - Field 63

''Contract Date - Field 65

''Current Maturity Date - Field 70

''Current Term - Field 72

''Scheduled Payment Amount - Field 74

''Closed Date - Field 76

''Payments Made - Field 78

''Current Payment Amount - Field 80

''Scheduled Due Date - Field 82

''Past Due Amount - Field 86

''Account Owner Name - Field 104

''Account street address - Field 107

''Account Owner City - Field 122

''Account Owner State - Field 123

''Account Owner Zip - Field 124

''Collateral Code - Field 126

''Description - Field 136

memberSSN = Trim(CurrentHost.Fields(27))

accountStatus = Trim(CurrentHost.Fields(29))

accountPayoffAmount = Trim(CurrentHost.Fields(33))

accountOriginalLoanAmount = Trim(CurrentHost.Fields(40))

accountCurrentRate = Trim(CurrentHost.Fields(42))

accountProductCode = Trim(CurrentHost.Fields(44))

accountOriginalProceeds = Trim(CurrentHost.Fields(46))

accountCurrentPrinciple = Trim(CurrentHost.Fields(63))

accountContractDate = Trim(CurrentHost.Fields(65))

accountCurrentMaturityDate = Trim(CurrentHost.Fields(70))

accountCurrentTerm = Trim(CurrentHost.Fields(72))

accountScheduledPaymentAmount = Trim(CurrentHost.Fields(74))

accountClosedDate = Trim(CurrentHost.Fields(76))

accountPaymentsMade = Trim(CurrentHost.Fields(78))

accountCurrentPaymentAmount = Trim(CurrentHost.Fields(80))

accountScheduledDueDate = Trim(CurrentHost.Fields(82))

accountPastDueAmount = Trim(CurrentHost.Fields(86))

memberName = Trim(CurrentHost.Fields(104))

memberStreetAddress = Trim(CurrentHost.Fields(107))

memberCityStateAddress = Trim(CurrentHost.Fields(122) & ", " & CurrentHost.Fields(123) & " " & CurrentHost.Fields(124))

accountCollateralCode = Trim(CurrentHost.Fields(126))

accountDescription = Trim(CurrentHost.Fields(136))

'Does not seem to have the issue of multiple account owners being listed

'trim ssn/TIN

memberSSN = Right(memberSSN, 9) 'remove leading digits on SSN/TIN

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

retrieveMemberAMAIInfo = False

End If

'THIS FUNCTION IS USED TO FIGURE OUT THE FIELD NUMBERS ARE FOR A HOST SCREEN================

'Dim i As Integer 'display the fields

'For i = 6 To CurrentHost.Fields.Count - 1

'column g is index, column h is value

' With Worksheets("AutoQuery").Range("G" & CStr(i))

' .Value = CStr(i)

' End With

' With Worksheets("AutoQuery").Range("H" & CStr(i))

' .Value = CStr(CurrentHost.Fields(i))

'End With

'Next i 'end of for loop

'===========================================================================================

End Function 'end of retrieveMemberAMAIInfo

Function displayMemberAMAIInfoResults(ByVal index As Integer, ByVal existingValue As Boolean, ByVal memberName As String, ByVal memberStreetAddress As String, ByVal memberCityStateAddress As String, ByVal memberSSN As String, ByVal accountProductCode As String, ByVal accountStatus As String, ByVal accountPayoffAmount As String, ByVal accountOriginalLoanAmount As String, ByVal accountCurrentRate As String, ByVal accountOriginalProceeds As String, ByVal accountCurrentPrinciple As String, ByVal accountContractDate As String, ByVal accountCurrentMaturityDate As String, ByVal accountCurrentTerm As String, ByVal accountScheduledPaymentAmount As String, ByVal accountClosedDate As String, ByVal accountPaymentsMade As String, ByVal accountCurrentPaymentAmount As String, ByVal accountScheduledDueDate As String, ByVal accountPastDueAmount As String, ByVal accountCollateralCode As String, ByVal accountDescription As String)

If (existingValue = True) Then

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

.Value = CStr(existingValue)

End With

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

.Value = memberName

End With

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

.Value = memberStreetAddress

End With

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

.Value = memberCityStateAddress

End With

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

.Value = memberSSN

End With

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

.Value = accountProductCode

End With

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

.Value = accountStatus

End With

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

.Value = accountPayoffAmount

End With

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

.Value = accountOriginalLoanAmount

End With

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

.Value = accountCurrentRate

End With

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

.Value = accountOriginalProceeds

End With

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

.Value = accountCurrentPrinciple

End With

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

.Value = accountContractDate

End With

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

.Value = accountCurrentMaturityDate

End With

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

.Value = accountCurrentTerm

End With

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

.Value = accountScheduledPaymentAmount

End With

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

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

.Value = "N/A"

Else

.Value = accountClosedDate

End If

End With

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

.Value = accountPaymentsMade

End With

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

.Value = accountCurrentPaymentAmount

End With

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

.Value = accountScheduledDueDate

End With

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

.Value = accountPastDueAmount

End With

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

.Value = accountCollateralCode

End With

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

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

.Value = "No description entered"

Else

.Value = accountDescription

End If

End With

Else 'if existing value is false

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

.Value = CStr(existingValue)

End With

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

.Value = "N/A"

End With

End If 'end of if existing value = true

End Function 'end of displayMemberAMAIInfoResults

Function getNumberOfAccountOwners(ByVal firstEntry As String, ByVal secondEntry As String, ByVal thirdEntry As String, ByVal fourthEntry As String, ByVal fifthEntry As String, ByVal sixthEntry As String) As Integer

Dim numOfOwners As Integer

Dim entrys(1 To 6) As String 'array of entrys

Dim Entry As Variant

numOfOwners = 0

entrys(1) = firstEntry

entrys(2) = secondEntry

entrys(3) = thirdEntry

entrys(4) = fourthEntry

entrys(5) = fifthEntry

entrys(6) = sixthEntry

For Each Entry In entrys

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

'if nothing, do nothing

Else

'increment counter

numOfOwners = numOfOwners + 1

End If

Next Entry

getNumberOfAccountOwners = (numOfOwners - 2) 'subtract 2 for address lines

End Function 'end of getNumberOfAccountOwners

Sub Auto\_Query\_Host\_Button()

'disable features to optimize macro for speed

Call disableFeaturesForMacroOptimization

'variable declaration

Dim finished As Boolean

Dim valueToQuery As String

Dim index As Integer

Dim valueType As String

Dim memberName As String

Dim memberStreetAddress As String

Dim memberCityStateAddress As String

Dim memberZipAddress As String

Dim memberBirthday As String

Dim memberCIS As String

Dim memberSSN As String

Dim accountProductCode As String

Dim accountType As String

Dim accountStatus As String

Dim accountPorB As String 'personal or commercial

Dim accountBalance As String

Dim accountMemoBalance As String 'for STI1

Dim accountCurrentBalance As String 'for STI1

Dim accountLoanBalance As String

Dim accountHolds As String 'for STI1

Dim accountPayoffAmount As String 'for AMAI

Dim accountOriginalLoanAmount As String 'for AMAI

Dim accountCurrentRate As String 'for AMAI

Dim accountOriginalProceeds As String 'for AMAI

Dim accountCurrentPrinciple As String 'for AMAI

Dim accountContractDate As String 'for AMAI

Dim accountCurrentMaturityDate As String 'for AMAI

Dim accountCurrentTerm As String 'for AMAI

Dim accountScheduledPaymentAmount As String 'for AMAI

Dim accountClosedDate As String 'for AMAI

Dim accountPaymentsMade As String 'for AMAI

Dim accountCurrentPaymentAmount As String 'for AMAI

Dim accountScheduledDueDate As String 'for AMAI

Dim accountPastDueAmount As String 'for AMAI

Dim accountCollateralCode As String 'for AMAI

Dim accountDescription As String 'for AMAI

'initialize variables---------------------------------------------

finished = False

memberName = ""

memberStreetAddress = ""

memberCityStateAddress = ""

memberZipAddress = ""

memberBirthday = ""

memberCIS = ""

memberSSN = ""

accountProductCode = ""

accountType = ""

accountStatus = ""

accountPorB = ""

accountBalance = ""

accountMemoBalance = ""

accountCurrentBalance = ""

accountLoanBalance = ""

accountHolds = ""

accountPayoffAmount = ""

accountOriginalLoanAmount = ""

accountCurrentRate = ""

accountOriginalProceeds = ""

accountCurrentPrinciple = ""

accountContractDate = ""

accountCurrentMaturityDate = ""

accountCurrentTerm = ""

accountScheduledPaymentAmount = ""

accountClosedDate = ""

accountPaymentsMade = ""

accountCurrentPaymentAmount = ""

accountScheduledDueDate = ""

accountPastDueAmount = ""

accountCollateralCode = ""

accountDescription = ""

index = 5 'start reading excel sheet on row 5

With Worksheets("AutoQuery").Range("A" & CStr(2))

valueType = Trim(.Value)

End With

'end of initialize variables---------------------------------------------

'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

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

'check to see if logged into host--------------------------------------

If (LoggedIntoHost = False) Then

MsgBox "ERROR: Host session has expired. Please Login to Host again."

Exit Sub

End If

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

'Make sure Value type is not blank------------------------

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

MsgBox "ERROR: Select a Value Type (Cell A2)"

Exit Sub

End If

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

'loop through entrys-------------------------------------------

Do While (finished = False)

'load value from excel

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

valueToQuery = Trim(.Value)

End With

'check to make sure value is not empty

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

finished = True

'display error message if no values are present to look up

If (index = 5) Then

MsgBox "ERROR: There must be at least one value in the Value's to Query Column. Make sure the first value is on row 5."

End If 'end of if index =5

Exit Do

'if not empty, query host using value

Else

'see if value exists

ResetForCommand

Dim existingValue As Boolean

'retrieve info, based upon valueType

If (StrComp(valueType, "IM Account Number") = 0) Then

Call displayMemberIMI1InfoHeaders

existingValue = retrieveMemberIMI1Info(valueToQuery, valueType, accountProductCode, accountType, accountStatus, accountPorB, accountBalance, accountLoanBalance, memberName, memberStreetAddress, memberCityStateAddress, memberSSN)

Call displayMemberIMI1InfoResults(index, existingValue, memberName, memberStreetAddress, memberCityStateAddress, memberSSN, accountProductCode, accountType, accountStatus, accountPorB, accountBalance, accountLoanBalance)

ElseIf (StrComp(valueType, "ST Account Number") = 0) Then

Call displayMemberSTI1InfoHeaders

existingValue = retrieveMemberSTI1Info(valueToQuery, valueType, accountProductCode, accountType, memberName, accountMemoBalance, memberStreetAddress, accountCurrentBalance, memberCityStateAddress, accountHolds, memberCIS, memberSSN, accountStatus, accountPorB)

Call displayMemberSTI1InfoResults(index, existingValue, memberName, memberStreetAddress, memberCityStateAddress, memberSSN, memberCIS, accountProductCode, accountType, accountStatus, accountPorB, accountMemoBalance, accountCurrentBalance, accountHolds)

ElseIf (StrComp(valueType, "AM Account Number") = 0) Then

Call displayMemberAMAIInfoHeaders

existingValue = retrieveMemberAMAIInfo(valueToQuery, valueType, memberName, memberStreetAddress, memberCityStateAddress, memberSSN, accountProductCode, accountStatus, accountPayoffAmount, accountOriginalLoanAmount, accountCurrentRate, accountOriginalProceeds, accountCurrentPrinciple, accountContractDate, accountCurrentMaturityDate, accountCurrentTerm, accountScheduledPaymentAmount, accountClosedDate, accountPaymentsMade, accountCurrentPaymentAmount, accountScheduledDueDate, accountPastDueAmount, accountCollateralCode, accountDescription)

Call displayMemberAMAIInfoResults(index, existingValue, memberName, memberStreetAddress, memberCityStateAddress, memberSSN, accountProductCode, accountStatus, accountPayoffAmount, accountOriginalLoanAmount, accountCurrentRate, accountOriginalProceeds, accountCurrentPrinciple, accountContractDate, accountCurrentMaturityDate, accountCurrentTerm, accountScheduledPaymentAmount, accountClosedDate, accountPaymentsMade, accountCurrentPaymentAmount, accountScheduledDueDate, accountPastDueAmount, accountCollateralCode, accountDescription)

Else 'if CIS or SSN/TIN

Call displayMemberRMI1InfoHeaders

existingValue = retrieveMemberRMI1Info(valueToQuery, valueType, memberName, memberSSN, memberCIS, memberBirthday, memberStreetAddress, memberCityStateAddress, memberZipAddress)

Call displayMemberRMI1InfoResults(index, existingValue, memberName, memberStreetAddress, memberCityStateAddress, memberZipAddress, memberBirthday, memberCIS, memberSSN)

End If 'end if valuetype is im account number

End If 'end of not the empty string

index = index + 1

Loop 'end of loop---------------------------------------------------------

're-enable features once macro is finished

Call enableFeaturesAfterOptimizedMacroIsDone

End Sub 'end of auto query host button