Option Explicit
' This module accesses the Data Tools product "Rapid Addressing Tool" to validate addresses.
' It uses late binding, so no reference is required.
' Note that it implicitly uses an Australian configuration.
'
' Record for passing values to, or returning values from, address validation.
Type AddressValidate
    sAddressLine As String ' The address line
    sLocality As String ' Suburb
    sState As String ' State
    iPostcode As Integer ' Postcode
    bValidated As Boolean ' Was the address successfully validated by Data Tools?
End Type
' Validate an address that has been entered.
Function ValidateAddress(recAddressIn As AddressValidate) As AddressValidate
    Const csTitle As String = "Address Validation"
    Dim oRat As Object
    Dim iResultsFound As Integer
    Dim iNdx As Integer
    Dim bStatus As Boolean
    Dim bAllBlank As Boolean
    Dim iAddressSelected As Integer
    Dim sLine As String
    ValidateAddress.bValidated = False
    Set oRat = CreateObject("DataTools_DtRat.Dt")
    ' Open Engine
    bStatus = oRat.DtOpen
    If Not bStatus Then
        MsgBox "Error validating address"
        Exit Function
    End If
    ' Are all of the input fields blank?
    bAllBlank = False
    With recAddressIn
        If IsBlank(.sAddressLine) Then
            If IsBlank(.sLocality) Then
                If IsBlank(.sState) Then
                    If (.iPostcode = 0) Then
                        bAllBlank = True
                    End If
                End If
            End If
        End If
        If bAllBlank Then
            iResultsFound = -1
        Else
            ' Search using such info as has been supplied.
            iResultsFound = oRat.DtIntelliSearchAddressLine(_
                .sAddressLine, _
                Trim(.sLocality & Space(1) & .sState & Space(1) & FmtPostcode(.iPostcode)))
        End If
    End With
    ' Check the results of the validation.
    Select Case iResultsFound
        Case -1
            ' No details entered.
MsgError "Please enter address details before validating"
Case 0
' No matches.
MsgInfo "No matching addresses found", csTitle
Case 1
' 1 match.
iAddressSelected = 1
Case Else
' Multiple matches - display all results.
ReDim vList(0 To (iResultsFound - 1))
For iNdx = 1 To iResultsFound
    sLine = oRat.DtExtractResultField("Address_Line", iNdx) & Space(1)
    sLine = Trim(sLine & oRat.DtExtractResultField("Locality", iNdx)) & Space(1)
    sLine = Trim(sLine & oRat.DtExtractResultField("State", iNdx)) & Space(1)
    sLine = sLine & oRat.DtExtractResultField("Postcode", iNdx)
    vList(iNdx - 1) = Trim(sLine)
Next
' Allow user to select one address.
With frmSelAddress
    .lbxAddresses.List = vList
    .lbxAddresses.ListIndex = 0
    .Show
    If .chkSelectionMade.Value Then
        iAddressSelected = .lbxAddresses.ListIndex + 1
    Else
        iResultsFound = 0
    End If
End With
End Select
' Pass selected details back.
If (iResultsFound >= 1) Then
' =1 Address returned
With ValidateAddress
    .bValidated = True
    .sAddressLine = oRat.DtExtractResultField("Address_Line", iAddressSelected)
    .sLocality = oRat.DtExtractResultField("Locality", iAddressSelected)
    .sState = oRat.DtExtractResultField("State", iAddressSelected)
    .iPostcode = StrToInt(oRat.DtExtractResultField("Postcode", iAddressSelected))
End With
End If
' Close Engine
bStatus = oRat.DtClose
End Function