Red Sticker Appendix F - VIN Decoder

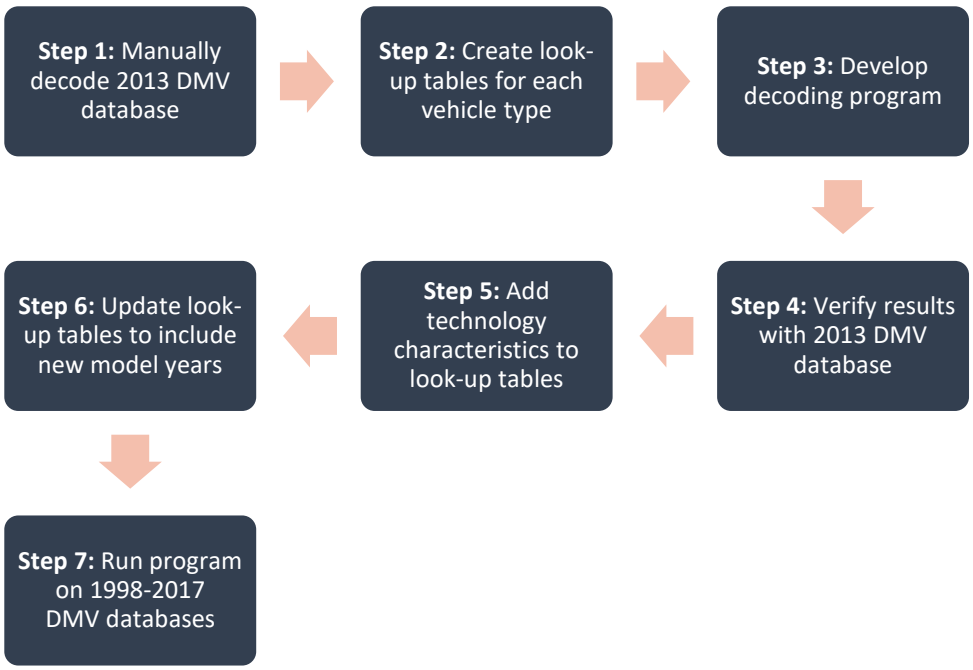
March 5, 2019
Monitoring and Laboratory Division

At the 2013 off-highway recreational vehicle (OHRV) rulemaking, the Board directed staff to conduct an assessment of the Red Sticker program and return with a comprehensive solution. The components of the assessment included a population evaluation, emissions testing, and an owner survey. To evaluate the population of the DMV database for OHRV, staff reviewed millions of DMV records and developed a new software program to update the population in the emissions inventory.

To accurately estimate emission from OHRV throughout California, it is critical to have a thorough understanding of the statewide OHRV population. It is important to consider the vehicle type (motorcycle, ATV, utility vehicle, etc.), vehicle age, engine technology (two-stroke, four-stroke, fuel injected, carbureted, etc.), engine power, and vehicle registration type (red sticker, green sticker, planned non-operation, etc.) when estimating emissions. The software previously used by CARB staff for evaluating OHRV population was the Polk vehicle identification number (VIN) Decoder. However, during this assessment staff found that the Polk VIN Decoder only identifies select OHRV models, and reports a high percentage of off-road vehicles as “unknown” because it was designed primarily for on-road vehicles. Furthermore, the Polk VIN Decoder does not include any information on OHRV engine or fuel delivery type.

To provide the most robust evaluation of California’s OHRV population, CARB staff developed an internal VIN Decoder as an alternative to the Polk VIN Decoder. The CARB VIN Decoder is a program that searches through lookup tables that list the first 10 digits of each VIN in California’s DMV registration database and matches the correct make, model, and vehicle characteristics that influence emissions. The lookup tables used by the VIN Decoder were built on hours of staff time searching through DMV databases and matching the VINs with the correct OHRV makes and models. Staff looked through millions of DMV records and carefully dissected the make and model of OHRVs, then searched online to find additional information about the attributes of each make and model. Figure 1 shows CARB process for developing and utilizing a VIN Decoder.

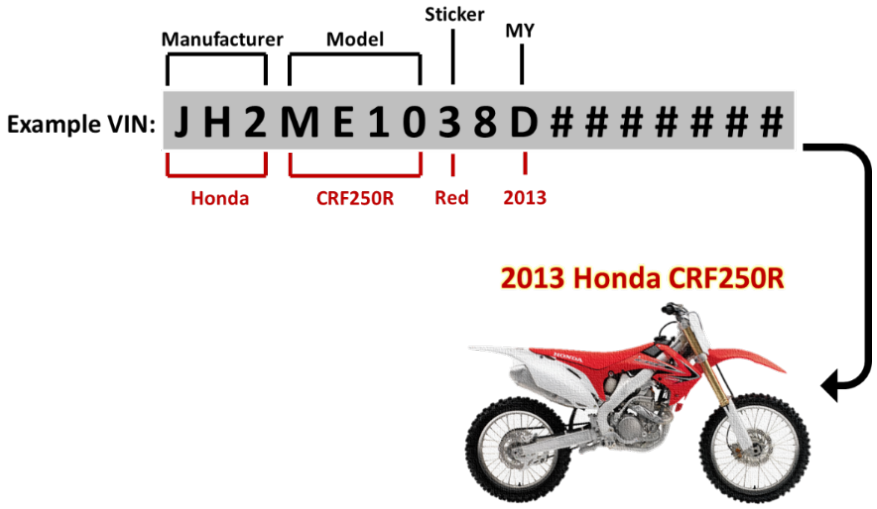
Figure 1: VIN Decoder Process



Step 1: Manually Decode DMV Database

The first step in using the VIN Decoder for evaluating the population was to manually decode a DMV database to match the VIN with the correct make and model. A diagram depicting how the 10 digits of the VIN were evaluated is shown in Figure 2.

Figure 2: Decoding of 10-Digit VIN



The first 3 digits of the VIN indicate the manufacturer of the OHRV. The next four digits represent the model. The eighth digit of the VIN was set to represent the sticker type of the OHRV indicating whether it was registered as a green or red sticker. Finally, the tenth digit was used to identify the model year of the vehicle. The remaining VIN digits do not represent any attributes of the OHRV.

Step 2: Create Lookup Tables

After staff identified the make and models and compared to the OHRVs online, look up tables were populated with the ten digit VIN and the corresponding OHRV attributes in the following columns as shown in Figure 3.

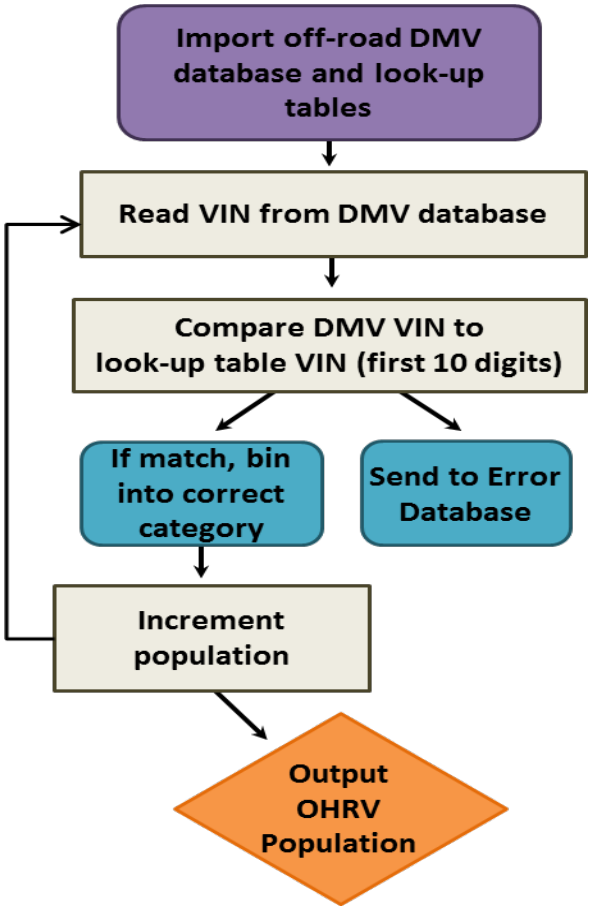
Figure 3: Example of Lookup Table

VIN	MAKE	MODEL	YEAR	STICKER
JYACE25Y-D	YAMA	TT-R110E	2013	GREEN
JYACE25Y-C	YAMA	TT-R110E	2012	GREEN
JYACE25T-B	YAMA	TT-R110E	2011	GREEN
JYACE25Y-B	YAMA	TT-R110E	2011	GREEN
JYACE25Y-9	YAMA	TT-R110E	2009	GREEN
JYACE25Y-8	YAMA	TT-R110E	2008	GREEN
JYACE25Y-5	YAMA	TT-R110E	2005	GREEN
JYACE07Y-6	YAMA	TTR125	2006	GREEN
JYACE07Y-5	YAMA	TTR125	2005	PRE03G
JYACE074-3	YAMA	TTR125	2003	GREEN
JYACE07T-3	YAMA	TTR125	2003	GREEN
JYACE071-2	YAMA	TTR125	2002	PRE03G
JYACE074-2	YAMA	TTR125	2002	PRE03G
JYACE07Y-1	YAMA	TTR125	2001	PRE03G
JYACE07W-Y	YAMA	TTR125	2000	PRE03G
JYACE07Y-Y	YAMA	TTR125	2000	PRE03G
JYACE114-A	YAMA	TTR125/LW	2010	GREEN
JYACE124-A	YAMA	TTR125/LW	2010	GREEN
JYACE07Y-4	YAMA	TTR125/LW	2004	GREEN
JYACE119-4	YAMA	TTR125/LW	2004	GREEN
JYACE11Y-4	YAMA	TTR125/LW	2004	GREEN

Step 3: Develop Decoder Program

After the lookup tables were populated, staff developed a program to match lookup tables to the 10-digit VINs that were in the DMV database. The flow chart for the process of matching VINs is presented in Figure 4.

Figure 4: ARB VIN Decoder Flowchart



The VIN Decoder was designed to import the DMV database and look-up tables into the program buffer to reduce search time. The VIN Decoder would then read each DMV row, one at a time, and compare the first ten digits to the ten digits in the lookup table. The VIN Decoder would then continue to look, row by row, to determine if there was a match. If it was not found, the VIN would be sent to an error database where staff could review it and update lookup tables as needed to include valid OHRV VINs. If a match was found, the program would increment the correct bin by type, year, etc. Once the VIN Decoder has looked through the complete DMV database, the program would output the information requested by the input parameters.

For clarification purposes, the VIN Decoder software designed for this rulemaking update doesn't actually decode VINs. The VIN Decoder software searches the DMV database for the first 10 digits of the VIN and then matches that to the lookup tables populated by staff. When it finds a match, it then increments the correct bin and moves onto the next file. The VINs are not decoded by the program itself.

Step 4: Verify Results

To confirm that the VIN Decoder was working properly, staff matched the results to known values for the 2013 DMV database. The comparison of our results to the known values are shown in Figure 5.

Figure 5: Comparison of VIN Decoder to Manual Decoding Results

	Manual Decoding		VIN Decoder		Difference	
	Active	Inactive	Active	Inactive	**Active**	**Inactive**
Green	158,665	71,431	158,619	71,386	**0.03%**	**0.06%**
Red	123,121	64,621	122,611	63,938	**0.41%**	**1.06%**

The comparison of results indicated that the VIN Decoder had found known values within approximately 1 percent of manually decoding the values. It is also interesting to note that the VIN Decoder, in general, had found more values than previous estimates. This is likely due to the increased accuracy of the program.

Step 5: Add Vehicle Attribute to the Lookup Tables

In order to evaluate vehicle attributes that only apply to particular models, staff searched online and populated the lookup table with additional information about the models. These vehicle attributes included technology type, displacement, fuel capacity, fuel delivery, and engine stroke. The tables were also developed to have drop in values where applicable to reduce error input. If a value was not found after searching, a "NF" value was given to ensure that the value was checked. An example of the lookup table with attributes is shown in Figure 6.

Figure 6: Lookup Table with Attributes

Vehicle Type						Drop-Down Menu				Fill-In				
VIN 10-digit	Man.	Make	Model Year	Sticker	Type	Fuel Delivery	Engine Stroke	Cylinders	Fuel Type	Engine HP	Displacement (cc)	Fuel Capacity (gal)	Dry Weight (lbs.)	MSRP
VBKMXA23-B	KTM	105 SX	2011	RED	OHMC	Carburetor	2 Stroke	1	Gas	NF	103.96	1.32	149.91	$ 5,499
VBKMXC23-A	KTM	105 SX	2010	RED	OHMC	Fuel Injection	2 Stroke	1	Gas	NF	103.96	1.32	149.91	$ 5,498
VBKMXC23-9	KTM	105 SX	2009	RED	OHMC	Carburetor	2 Stroke	1	Gas	NF	103.96	1.32	149.91	$ 5,498
VBKMXC23-8	KTM	105 SX	2008	RED	OHMC	NF	2 Stroke	1	Gas	NF	103.96	1.32	149.91	$ 5,498
VBKMXC23-7	KTM	105 SX	2007	RED	OHMC	Carburetor	2 Stroke	1	Gas	NF	104.9	1.3	149.9	$ 4,598
VBKMXC23-6	KTM	105 SX	2006	RED	OHMC	Carburetor	2 Stroke	1	Gas	NF	104.9	1.3	149.9	NF
VBKMXC23-5	KTM	105 SX	2005	RED	OHMC	Carburetor	2 Stroke	1	Gas	NF	104.9	1.3	149.9	NF
VBKMXC23-4	KTM	105 SX	2004	RED	OHMC	Carburetor	2 Stroke	1	Gas	NF	104.9	1.3	149.9	$ 3,798

Figure 7: Sample Output

VIN*	Manufacturer	Model	Model Year	Status	vehicle type	technology type	induction	Sticker type	Engine Size (cc)	Horse Power	Fuel Tank Size (g)	Fuel Type	Zip Code	County Code
V1	Honda	TRX500FA	2009	Active	ATV	4 stroke	Carburetor	GREEN	499	NF	4.1	Gas	95363	24
V2	Kawasaki	KLX450	2010	Active	OHMC	4 stroke	Fuel Injection	RED	249	58.19	2.2	Gas	94550	39
V3	Suzi	RM125	2002	Inactive	OHMC	2 stroke	Carburetor	RED	124.8	41	2.1	Gas	93274	40
V4	BOMB	DS 250	2012	Active	ATV	4 stroke	Carburetor	GREEN	249.4	NF	3.3	Gas	95949	41

* - Anonymous VINs

green shading - refers to data extracted from look up tables

blue shading - refers to data extracted from DMV database

Step 6: Update Lookup-Tables to Include Additional Model Years

To keep the VIN Decoder up to date, staff will need to include the latest model years and their attributes for ongoing years. Currently, the VIN Decoder only includes model years up to 2017.

Step 7: Run Program on all DMV Databases

Finally, once all the models have been updated to the current model year, staff can run the program for additional model years to determine the latest population results. A sample of the output is shown in Figure 7 on the previous page.

VIN Decoder Issues

Although robust in development and research, the VIN Decoder is only as good as the information provided in the DMV database. The VIN Decoder relies on the correctness of the DMV database and the values that have been input by the DMV. The DMV database includes a field to identify vehicle body type, but staff noticed categorization of vehicle types were not correctly input in many cases, particularly with small volume manufacturers. In addition, Chinese-manufactured vehicles commonly had inconsistent VINs, making those vehicles difficult to classify accurately. Regardless of these issues, the CARB VIN Decoder analysis of DMV registration data provides the most robust way to evaluate California's OHRV population.

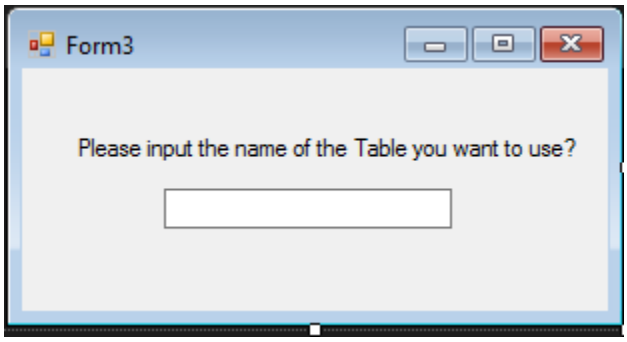
VIN Decoder Functionality

1. Main User Interface

When the VIN Decoder executable is run, the main user interface opens (see below). From here the user can run the model and monitor the status of its progress.

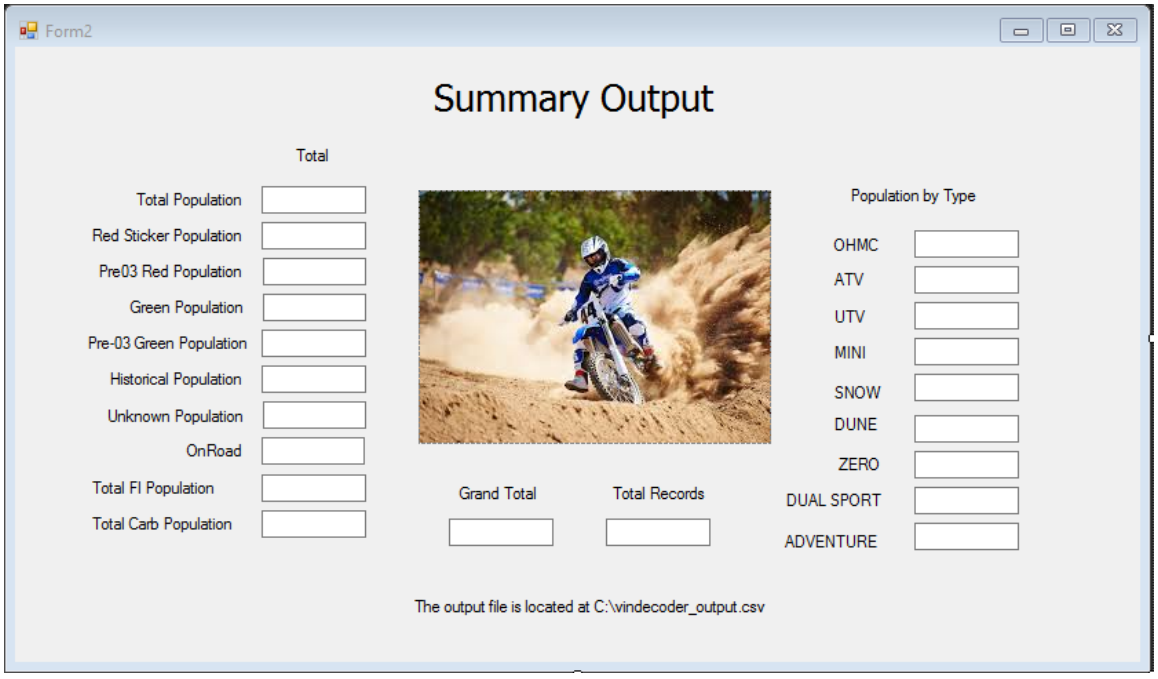
2. Search Table

Once the Run button has been selected, the user will then be asked to find a table to search through.



3. Summary Output

After the VIN Decoder is complete, a summary table will be displayed and an excel output (Figure 7) will be put in the c:\temp folder for further evaluation.



Attachment 1: Source code of VIN Decoder

```
Imports System
Imports System.Data
Imports System.Data.OleDb
Imports System.IO


Public Class Form1
    Dim provider As String
    Dim dataFile As String
    Dim connString As String
    Dim count As Integer
    Dim VIN As String = "vin"
    Dim VIN2 As String
    Dim VIN3 As String
    Dim ErrorVIN As String

    Public myConnection As OleDbConnection = New OleDbConnection
    Public dr As OleDbDataReader
    Public dr2 As OleDbDataReader
    Public dr3 As OleDbDataReader
    Public WMI As OleDbDataReader
    Public WMI2 As OleDbDataReader
    Public WMI3 As OleDbDataReader
    Public WMI4 As OleDbDataReader
    Public WMI5 As OleDbDataReader
    Public WMI6 As OleDbDataReader
    Public WMI7 As OleDbDataReader
    Public WMI8 As OleDbDataReader

    Private COL2 As Integer

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        myConnection.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\VBpractice\ohrv2.accdb"
    End Sub



    Private Sub RunButton_Click(sender As Object, e As EventArgs) Handles RunButton.Click

        'set up instances to be used in program
            Dim greenstickerPop, redstickerPop, histstickerPop, unknownPop, errorcheck, i, j, k, m,
            rs3pop, gs3pop, status, count2, ohmc, atv, utv, mini, snow, dune, dualsport, twoStroke,
            fourStroke, grandTotal, found As Integer

            Dim carb, fi, zero, ds As Integer

            Dim sticker, TestVIN10, TestVIN, VINString, myString, make, code, LC, make2, make3,
            model2, year2, stroke, ohrvType, license, fueldelivery, xactive, horsepower, zipcode,
            enginesize, mileage, fueltank, fueltype, countycode As String

            Dim TopMan1, TopMan2, TopMan3, TopMan4, TopMan5, onroad, adventure As Integer

            Dim ZCTA, CO, AB, DIS, GAI, CO_Name, year As String
```

```
        Dim RedActive, GreenActive, RedInactive, GreenInactive, HistActive, HistInactive,
        PRedActive, PGreenActive, PRedInactive, PGreenInactive, TotalActive, TotalInactive, x As
        Integer

    Dim StatusCode As Char

        greenstickerPop = 0
        redstickerPop = 0
        unknownPop = 0
        rs3pop = 0
        gs3pop = 0

        count = 0

        'Find location of database to be used
        MsgBox("Please find the location of the file")
        Dim ofd As OpenFileDialog = New OpenFileDialog

        If ofd.ShowDialog = DialogResult.OK Then
            'MsgBox(ofd.FileName)
        End If

        'Create and Open Connection from DMV database
        Dim conIeam As New OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & ofd.FileName & "")
        conIeam.Open()

        Dim conIeam2 As New OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\temp\output.accdb")
        conIeam2.Open()

        Dim inputData As String
        inputData = InputBox("Please input the name of Table:", "Input Box Text")

        Dim commdata As New OleDb.OleDbCommand("SELECT VIN, County, ZIP, Status, License
FROM " & inputData & "", conIeam)
        'Dim commdata2 As New OleDb.OleDbCommand("Select Cost FROM Items", conIeam)
        Dim usernameParam As New OleDb.OleDbParameter("@CodeText", VIN)
        commdata.Parameters.Add(usernameParam)
        dr = commdata.ExecuteReader()

        Dim rows As Integer
        Dim commdata2 As New OleDb.OleDbCommand("SELECT Count(*) FROM " & inputData & "",
conIeam)
        rows = commdata2.ExecuteScalar()
        'Table 1 is created for all ten digit lookup

        ProgressBar1.Minimum = 0
        ProgressBar1.Maximum = rows '2648752 number of total records in DMV database
        ProgressBar1.Value = 0

        'Table 2 is created for less than four digit lookup
        Dim cmd7 As New OleDbCommand("Select FIRST4, MODEL, year, make, sticker, Type FROM
VIN4", conIeam)
        WMI7 = cmd7.ExecuteReader()
        Dim table7 As New DataTable()
        table7.Load(WMI7)
```

```
        Dim cmd As New OleDbCommand("Select VIN, MANUFACTURER, MODELYEAR, make,
sticker, Type, ENGINEDELIVERY, fueldelivery, fueltype, DISPLACEMENT, FUELCAPACITY,
enginesize FROM OAZ", conIeam)
        WMI = cmd.ExecuteReader()
        Dim table As New DataTable()
        table.Load(WMI)

        Dim cmd2 As New OleDbCommand("Select VIN, MANUFACTURER, MODELYEAR, make,
sticker, Type, ENGINEDELIVERY, fueldelivery, fueltype, DISPLACEMENT, FUELCAPACITY,
enginesize FROM BCZ", conIeam)
        WMI2 = cmd2.ExecuteReader()
        Dim table2 As New DataTable()
        table2.Load(WMI2)

        Dim cmd3 As New OleDbCommand("Select VIN, MANUFACTURER, MODELYEAR, make,
sticker, Type, ENGINEDELIVERY, fueldelivery, fueltype, DISPLACEMENT, FUELCAPACITY,
enginesize FROM DJZ", conIeam)
        WMI3 = cmd3.ExecuteReader()
        Dim table3 As New DataTable()
        table3.Load(WMI3)

        Dim cmd4 As New OleDbCommand("Select VIN, MANUFACTURER, MODELYEAR, make,
sticker, Type, ENGINEDELIVERY, fueldelivery, fueltype, DISPLACEMENT, FUELCAPACITY,
enginesize FROM KRZ", conIeam)
        WMI4 = cmd4.ExecuteReader()
        Dim table4 As New DataTable()
        table4.Load(WMI4)

        Dim cmd5 As New OleDbCommand("Select VIN, MANUFACTURER, MODELYEAR, make,
sticker, Type, ENGINEDELIVERY, fueldelivery, fueltype, DISPLACEMENT, FUELCAPACITY,
enginesize FROM SZZ", conIeam)
        WMI5 = cmd5.ExecuteReader()
        Dim table5 As New DataTable()
        table5.Load(WMI5)


        Dim cmd6 As New OleDbCommand("Select ZCTA, CO, AB, DIS, GAI, CO_Name FROM GAI",
conIeam)
        WMI6 = cmd6.ExecuteReader()
        Dim table6 As New DataTable()
        table6.Load(WMI6)



        Dim cmd14 As New OleDbCommand("Delete * FROM DUALSPORT", conIeam2)
        WMI8 = cmd14.ExecuteReader()

        ' Dim table6 As New DataTable()

        Dim Num1 As Double
        'Label7.Text = TimeOfDay.ToString("h: mm:ss tt")

        'Table 3 is created array for storing data table and flushing to Excel
        Dim table8 As New DataTable()

        'Each BArray is big enough to hold enough information for an Excel sheet
        Dim BArray(1000000, 7) As String
        Dim BArray2(1000000, 7) As String
        Dim BArray3(1000000, 7) As String
```

```
Dim BArray4(1000000, 7) As String
Dim BArray5(1000000, 7) As String
Dim BArray6(1000000, 7) As String


'initialize variables
mini = 0
ohmc = 0
atv = 0
utv = 0
snow = 0
dune = 0

twoStroke = 0
fourStroke = 0
carb = 0
fi = 0
zero = 0

'overall total of found records
grandTotal = 0
year = 0
count2 = 0
count = 0
adventure = 0
'Start reading each row of DMV file, one by one
If dr.HasRows Then
   While dr.Read()
      'Increment status bar each time it reads a row
      ProgressBar1.Value += 1


      'Display count of the total DMV records
      Num1 = (count / (rows - 1))
      Label7.Text = Num1.ToString("00.00" + "%")

      Application.DoEvents()
      license = ""
      zipcode = ""
      'Read in single VIN
      count = count + 1

      'Store DMV values in new variables

      VIN2 = dr(0).ToString
      VIN3 = dr(0).ToString
      countycode = dr(1).ToString
      zipcode = dr(2).ToString
      StatusCode = dr(3).ToString
      license = dr(4).ToString
      'make2 = dr(5).ToString
      'make3 = dr(5).ToString
      'model2 = dr(6).ToString
      ' year2 = dr(7).ToString
      'Initializing variables
      status = 0
      errorcheck = 0
      ohrvType = ""
      stroke = ""
```

```
            fueldelivery = ""
            enginesize = 0
            fueltype = 0
            fueltank = 0
            horsepower = 0
            mileage = 0

            'Convert VIN to chars in an array
            Dim myChars() As Char = VIN2.ToCharArray
            x = 0
            'Initialize check to confirm value was found in the DMV database
            k = 0
            found = 0

            'confirm length of ten before adding adding VIN to 10 chars
            'set TestVIN10 to either three or ten characters
            If myChars.Length > 9 Then
               TestVIN10 = (myChars(0) + myChars(1) + myChars(2) + myChars(3) + myChars(4) +
myChars(5) + myChars(6) + myChars(7) + "-" + myChars(9))

               grandTotal = grandTotal + 1
            ElseIf myChars.Length > 3 And myChars.Length < 10 Then
               TestVIN10 = (myChars(0) + myChars(1) + myChars(2) + myChars(3))
               x = 7
               grandTotal = grandTotal + 1
            Else
               TestVIN10 = ""
            End If

            'read in WMI and Sticker codes and evaluate them

            '(re)initialize variables
            code = 0
            sticker = 0
            make = 0

            i = -1
            j = 0
            m = 0

            Select Case myChars(0)
               Case "0" To "A"
                  x = 1
               Case “B” To “C”
                  x = 2
               Case “D” To "J"
                  x = 3
               Case “K” To “R”
                  x = 4
               Case “S” To “Z”
                  x = 5
            End Select



            'use correct table for reading info
            'if x = 1 then compare the VIN10 tbale
            'if x = 2 compare to teh VIN4 table
            If x = 1 Then
```

```
            For Each row As DataRow In table.Rows
                i = i + 1
                'if found then move values to variables, otherwise iterate through each element of the
table
                If (StrComp(TestVIN10, table.Rows(i)(j + 0), CompareMethod.Text) = 0) Then
                    code = table.Rows(i)(j + 1)
                    year = table.Rows(i)(j + 2).ToString
                    make = table.Rows(i)(j + 3)
                    sticker = table.Rows(i)(j + 4)
                    ohrvType = table.Rows(i)(j + 5)
                    stroke = table.Rows(i)(j + 6)
                    fueldelivery = table.Rows(i)(j + 7).ToString
                    enginesize = table.Rows(i)(j + 9).ToString
                    horsepower = table.Rows(i)(j + 11).ToString
                    fueltank = table.Rows(i)(j + 10).ToString
                    fueltype = table.Rows(i)(j + 8).ToString

                    k = 1
                    count2 = count2 + 1
                    found = 1
                    Exit For
                End If

            Next
        ElseIf x = 2 Then
            For Each row As DataRow In table2.Rows
                i = i + 1
                'MsgBox(table2.Rows(i)(j + 0))
                If (StrComp(TestVIN10, table2.Rows(i)(j + 0), CompareMethod.Text) = 0) Then
                    code = table2.Rows(i)(j + 1)
                    year = table2.Rows(i)(j + 2).ToString
                    make = table2.Rows(i)(j + 3).ToString
                    sticker = table2.Rows(i)(j + 4).ToString
                    ohrvType = table2.Rows(i)(j + 5).ToString
                    stroke = table2.Rows(i)(j + 6).ToString
                    fueldelivery = table2.Rows(i)(j + 7).ToString
                    enginesize = table2.Rows(i)(j + 9).ToString
                    horsepower = table2.Rows(i)(j + 11).ToString
                    fueltank = table2.Rows(i)(j + 10).ToString
                    fueltype = table2.Rows(i)(j + 8).ToString
                    k = 1
                    count2 = count2 + 1
                    found = 1
                    Exit For
                End If
            Next
        ElseIf x = 3 Then
            For Each row As DataRow In table3.Rows
                i = i + 1
                'MsgBox(table2.Rows(i)(j + 0))
                If (StrComp(TestVIN10, table3.Rows(i)(j + 0), CompareMethod.Text) = 0) Then
                    code = table3.Rows(i)(j + 1)
                    year = table3.Rows(i)(j + 2).ToString
                    make = table3.Rows(i)(j + 3).ToString
                    sticker = table3.Rows(i)(j + 4).ToString
                    ohrvType = table3.Rows(i)(j + 5).ToString
                    stroke = table3.Rows(i)(j + 6).ToString
                    fueldelivery = table3.Rows(i)(j + 7).ToString
                    enginesize = table3.Rows(i)(j + 9).ToString
```

```
            horsepower = table3.Rows(i)(j + 11).ToString
            fueltank = table3.Rows(i)(j + 10).ToString
            fueltype = table3.Rows(i)(j + 8).ToString
            k = 1
            count2 = count2 + 1
            found = 1
            Exit For
        End If
    Next
ElseIf x = 4 Then
    For Each row As DataRow In table4.Rows
        i = i + 1
        'MsgBox(table2.Rows(i)(j + 0))
        If (StrComp(TestVIN10, table4.Rows(i)(j + 0), CompareMethod.Text) = 0) Then
            code = table4.Rows(i)(j + 1)
            year = table4.Rows(i)(j + 2).ToString
            make = table4.Rows(i)(j + 3).ToString
            sticker = table4.Rows(i)(j + 4).ToString
            ohrvType = table4.Rows(i)(j + 5).ToString
            stroke = table4.Rows(i)(j + 6).ToString
            fueldelivery = table4.Rows(i)(j + 7).ToString
            enginesize = table4.Rows(i)(j + 9).ToString
            horsepower = table4.Rows(i)(j + 11).ToString
            fueltank = table4.Rows(i)(j + 10).ToString
            fueltype = table4.Rows(i)(j + 8).ToString
            k = 1
            count2 = count2 + 1
            found = 1
            Exit For
        End If
    Next
ElseIf x = 5 Then
    For Each row As DataRow In table5.Rows
        i = i + 1
        'MsgBox(table2.Rows(i)(j + 0))
        If (StrComp(TestVIN10, table5.Rows(i)(j + 0), CompareMethod.Text) = 0) Then
            code = table5.Rows(i)(j + 1)
            year = table5.Rows(i)(j + 2).ToString
            make = table5.Rows(i)(j + 3).ToString
            sticker = table5.Rows(i)(j + 4).ToString
            ohrvType = table5.Rows(i)(j + 5).ToString
            stroke = table5.Rows(i)(j + 6).ToString
            fueldelivery = table5.Rows(i)(j + 7).ToString
            enginesize = table5.Rows(i)(j + 9).ToString
            horsepower = table5.Rows(i)(j + 11).ToString
            fueltank = table5.Rows(i)(j + 10).ToString
            fueltype = table5.Rows(i)(j + 8).ToString
            k = 1
            count2 = count2 + 1
            found = 1
            Exit For
        End If
    Next

ElseIf x = 7 Then
    For Each row As DataRow In table7.Rows
        i = i + 1
        'MsgBox(table2.Rows(i)(j + 0))
        If (StrComp(TestVIN10, table7.Rows(i)(j + 0), CompareMethod.Text) = 0) Then
```

```
            code = table7.Rows(i)(j + 1)
            year = table7.Rows(i)(j + 2).ToString
            make = table7.Rows(i)(j + 3).ToString
            sticker = table7.Rows(i)(j + 4).ToString
            ohrvType = table7.Rows(i)(j + 5).ToString
            stroke = table7.Rows(i)(j + 6).ToString
            fueldelivery = table7.Rows(i)(j + 7).ToString
            enginesize = table7.Rows(i)(j + 9).ToString
            horsepower = table7.Rows(i)(j + 11).ToString
            fueltank = table7.Rows(i)(j + 10).ToString
            fueltype = table7.Rows(i)(j + 8).ToString
            k = 1
            count2 = count2 + 1
            found = 1
            Exit For
        End If
    Next
End If

'Find GAI information

For Each row As DataRow In table6.Rows

    'MsgBox(table2.Rows(i)(j + 0))

    If (StrComp(zipcode, table6.Rows(m)(j + 0), CompareMethod.Text) = 0) Then
        ZCTA = table6.Rows(m)(j + 0)
        CO = table6.Rows(m)(j + 1)
        AB = table6.Rows(m)(j + 2)
        DIS = table6.Rows(m)(j + 3)
        GAI = table6.Rows(m)(j + 4)
        CO_Name = table6.Rows(m)(j + 5)
        Exit For
    End If
    m = m + 1
Next


'Set a status code for inactive/active each row
Select Case StatusCode
    Case "C"
        status = 1
    Case "E"
        status = 1
    Case "S"
        status = 1
    Case "N"
        status = 0
    Case "P"
        status = 0
    Case "R"
        status = 0
End Select
'Move elements of datatable into array for excel dump
'Separate into each excel spreadsheet

If status = 1 Then
    xactive = "active"
ElseIf status = 0 Then
```

```
    xactive = "inactive"
Else
    xactive = ""
End If

If found = 1 Then
    If count2 <= 1000000 Then
        BArray(count2 - 1, 0) = VIN2
        BArray(count2 - 1, 1) = code
        BArray(count2 - 1, 2) = make
        BArray(count2 - 1, 3) = year
        BArray(count2 - 1, 4) = xactive
        BArray(count2 - 1, 5) = ohrvType

        BArray(count2 - 1, 6) = stroke


        BArray2(count2 - 1, 0) = fueldelivery

        BArray2(count2 - 1, 1) = sticker
        BArray2(count2 - 1, 2) = enginesize

        BArray2(count2 - 1, 3) = horsepower
        BArray2(count2 - 1, 4) = mileage
        BArray2(count2 - 1, 5) = fueltank
        BArray2(count2 - 1, 6) = fueltype


        BArray5(count2 - 1, 0) = zipcode


        BArray5(count2 - 1, 1) = CO

        BArray5(count2 - 1, 2) = AB
        BArray5(count2 - 1, 3) = DIS
        BArray5(count2 - 1, 4) = GAI
        BArray5(count2 - 1, 5) = CO_Name


    ElseIf count2 > 1000000 Then
        BArray3(count2 - 1000000, 0) = VIN2
        BArray3(count2 - 1000000, 1) = code
        BArray3(count2 - 1000000, 2) = make
        BArray3(count2 - 1000000, 3) = year
        BArray3(count2 - 1000000, 4) = xactive
        BArray3(count2 - 1000000, 5) = ohrvType

        BArray3(count2 - 1000000, 6) = stroke


        BArray4(count2 - 1000000, 0) = fueldelivery

        BArray4(count2 - 1000000, 1) = sticker
        BArray4(count2 - 1000000, 2) = enginesize

        BArray4(count2 - 1000000, 3) = horsepower
        BArray4(count2 - 1000000, 4) = mileage
        BArray4(count2 - 1000000, 5) = fueltank
        BArray4(count2 - 1000000, 6) = fueltype
```

```
        BArray6(count2 - 1000000, 0) = zipcode
        BArray6(count2 - 1000000, 1) = CO


        BArray6(count2 - 1000000, 2) = AB
        BArray6(count2 - 1000000, 3) = DIS
        BArray6(count2 - 1000000, 4) = GAI
        BArray6(count2 - 1000000, 5) = CO_Name

    End If
End If


If make2 = "ZERO" Then
    zero = zero + 1
End If

'Bin top 5 manufacturers
Select Case make
    Case "HOND"
        TopMan1 = TopMan1 + 1
    Case "YAMA"
        TopMan2 = TopMan2 + 1
    Case "KAWK"
        TopMan3 = TopMan3 + 1
    Case "KTM"
        TopMan4 = TopMan4 + 1
    Case "SUZI"
        TopMan5 = TopMan5 + 1
End Select

'Count engine delivery

Select Case stroke
    Case "2 Stroke"
        twoStroke = twoStroke + 1

    Case "4 Stroke"
        fourStroke = fourStroke + 1

End Select

Select Case fueldelivery
    Case "Carburetor"
        carb = carb + 1
    Case "Fuel Injection"
        fi = fi + 1
End Select


'Bin total population by type

Select Case ohrvType
    Case "OHMC"
        ohmc = ohmc + 1
    Case "MINI"
        mini = mini + 1
```

```
            Case "ATV"
                atv = atv + 1
            Case "UTV"
                utv = utv + 1
            Case "SNOW"
                snow = snow + 1
            Case "DUNE/SC"
                dune = dune + 1
            Case "DUAL SPORT"
                dualsport = dualsport + 1
            Case "ADVENTURE"
                adventure = adventure + 1
        End Select

        'Count only OHMC green and red stickers

        If sticker = "GREEN" Then
            greenstickerPop = greenstickerPop + 1

            code = 1

        ElseIf sticker = "RED" Then

            redstickerPop = redstickerPop + 1
            code = 1
        ElseIf sticker = "PRE03R" Then
            rs3pop = rs3pop + 1
            code = 1
        ElseIf sticker = "PRE03G" Then
            gs3pop = gs3pop + 1
            code = 1
        ElseIf sticker = "HIST" Then
            histstickerPop = histstickerPop + 1
        End If


        'if found in lookup table and found in license code as 21 then put into database table

        'if Dual sport, check to see if in database. If so, decrease duplicate by one, else place into
database
        ds = 0

        If ohrvType = "DUAL SPORT" Then

            Dim cmd11 As New OleDbCommand("SELECT * FROM DUALSPORT Where VIN = '" &
VIN3 & "' ", conIeam2)
            dr3 = cmd11.ExecuteReader
            dr3.Read()


            If dr3.HasRows Then
                dualsport = dualsport - 1
                ds = 1
            End If

            If ds = 0 Then
```

```
                    Dim cmd12 As New OleDbCommand("INSERT INTO
DUALSPORT(VIN,MAKE,MODEL,MODELYEAR) VALUES ('" & VIN3 & "', '" & code & "' , '" & make &
"', '" & year & "')", conIeam2)
                    cmd12.ExecuteNonQuery()

                End If
                dr3.Close()
            End If


            '  If k = 0 And license = "21" Then

            '    Dim cmd8 As New OleDbCommand("INSERT INTO Onroad(VIN,MAKE,MODEL,YEARS)
VALUES ('" & VIN2 & "', '" & code & "' , '" & make & "', '" & year & "')", conIeam2)
            '   cmd8.ExecuteNonQuery()
            onroad = onroad + 1

            '  ElseIf k = 0 Then

            If k = 0 Then

                'Dim cmd9 As New OleDbCommand("INSERT INTO
ErrorList(VIN,MAKE,MODEL,YEARS) VALUES ('" & VIN2 & "', '" & make3 & "' , '" & model2 & "', '" &
year & "')", conIeam2)
                ' cmd9.ExecuteNonQuery()

                unknownPop = unknownPop + 1
            End If

        End While

    End If


    'Move array into excel
    '########################################

    Dim oExcel As Object
    Dim oBook As Object
    Dim oSheet As Object


    'Start a new workbook in Excel
    oExcel = CreateObject("Excel.Application")
    oBook = oExcel.Workbooks.Add

    oSheet = oBook.sheets("sheet1")


    'Add headers to the worksheet on row 1

    oSheet.Range("A1").Value = ("VIN")
    oSheet.Range("B1").Value = ("Manufacturer")
    oSheet.Range("C1").Value = ("Model")
    oSheet.Range("D1").Value = ("Model Year")
    oSheet.Range("E1").Value = ("Status")
    oSheet.Range("F1").Value = ("Vehicle Type")
    oSheet.Range("G1").Value = ("Technology Type")
    oSheet.Range("H1").Value = ("Induction")
```

```
oSheet.Range("I1").Value = ("Sticker Type")
oSheet.Range("J1").Value = ("Engine Size (cc)")
oSheet.Range("K1").Value = ("Horsepower")
oSheet.Range("L1").Value = ("Mileage")
oSheet.Range("M1").Value = ("Fuel Tank Size (g)")
oSheet.Range("N1").Value = ("Fuel Type")
oSheet.Range("O1").Value = ("Zip Code")
oSheet.Range("P1").Value = ("County Code")

oSheet.Range("Q1").Value = ("Air Basin")
oSheet.Range("R1").Value = ("Air District")
oSheet.Range("S1").Value = ("GAI")
oSheet.Range("T1").Value = ("County Name")



'Transfer the array to the worksheet starting at cell A2

oSheet.Range("A2").Resize(1000100, 7).Value = BArray
oSheet.Range("A1:G1").EntireColumn.AutoFit()

' oSheet2.Select()

oSheet.Range("H2").Resize(1000100, 7).Value = BArray2
oSheet.Range("H1:N1").EntireColumn.AutoFit()

oSheet.Range("O2").Resize(1000100, 6).Value = BArray5
oSheet.Range("O1:T1").EntireColumn.AutoFit()

'oSheet.Range("A1040104").Resize(1040100, 8).Value = BArray
'oSheet.Range("A1:H1").EntireColumn.AutoFit()
' oSheet2.Select()

'oSheet.Range("I1040104").Resize(1040100, 8).Value = BArray2
'oSheet.Range("I1:P1").EntireColumn.AutoFit()

'Save the Workbook and Quit Excel

oBook.SaveAs("C:\temp\vindecoder_output.csv", 6)
oBook.Close(savechanges:=False)


If count2 > 1000000 Then


    Dim oBook2 As Object
    Dim oSheet2 As Object
    oBook2 = oExcel.Workbooks.Add

    oSheet2 = oBook2.sheets("sheet1")

    'Add headers to the worksheet on row 1

    oSheet2.Range("A1").Value = ("VIN")
    oSheet2.Range("B1").Value = ("Manufacturer")
    oSheet2.Range("C1").Value = ("Model")
    oSheet2.Range("D1").Value = ("Model Year")
    oSheet2.Range("E1").Value = ("Status")
    oSheet2.Range("F1").Value = ("Vehicle Type")
```

```
    oSheet2.Range("G1").Value = ("Technology Type")
    oSheet2.Range("H1").Value = ("Induction")
    oSheet2.Range("I1").Value = ("Sticker Type")
    oSheet2.Range("J1").Value = ("Engine Size (cc)")
    oSheet2.Range("K1").Value = ("Horsepower")
    oSheet2.Range("L1").Value = ("Mileage")
    oSheet2.Range("M1").Value = ("Fuel Tank Size (g)")
    oSheet2.Range("N1").Value = ("Fuel Type")
    oSheet2.Range("O1").Value = ("Zip Code")
    oSheet2.Range("P1").Value = ("County Code")

    oSheet2.Range("Q1").Value = ("Air Basin")
    oSheet2.Range("R1").Value = ("Air District")
    oSheet2.Range("S1").Value = ("GAI")
    oSheet2.Range("T1").Value = ("County Name")


    'Transfer the array to the worksheet starting at cell A2



    oSheet2.Range("A2").Resize(1000100, 7).Value = BArray3
    oSheet2.Range("A1:G1").EntireColumn.AutoFit()


    oSheet2.Range("H2").Resize(1000100, 7).Value = BArray4
    oSheet2.Range("H1:N1").EntireColumn.AutoFit()

    oSheet2.Range("O2").Resize(1000100, 6).Value = BArray6
    oSheet2.Range("O1:T1").EntireColumn.AutoFit()



    'oSheet.Range("I1040104").Resize(1040100, 8).Value = BArray2
    'oSheet.Range("I1:P1").EntireColumn.AutoFit()

    'Save the Workbook and Quit Excel


    oBook2.SaveAs("C:\temp\vindecoder_output2.csv", 6)
    oBook2.Close(savechanges:=False)

End If

oExcel.Quit

'#######################################



'Update form box with population values
Form2.Show()
Form2.TotalTextBox.Text = greenstickerPop + redstickerPop + rs3pop + gs3pop + histstickerPop

Form2.ohmc.Text = ohmc
Form2.atv.Text = atv
Form2.utv.Text = utv
Form2.mini.Text = mini
Form2.snow.Text = snow
```

```
        Form2.dune.Text = dune
        Form2.zero.Text = zero
        Form2.dualsport.Text = dualsport
        Form2.adventure.Text = adventure


        Form2.grandTotal.Text = grandTotal
        Form2.count.Text = count2

        Form2.FITextBox.Text = fi
        Form2.CARBTextBox.Text = carb


        Form2.HistTextBox.Text = histstickerPop
        Form2.UnkTextBox.Text = unknownPop
        Form2.onroad.Text = onroad
        Form2.RS3Pop.Text = rs3pop
        Form2.GS3Pop.Text = gs3pop
        Form2.RSPopTextBox.Text = redstickerPop

        Form2.GreenTextBox.Text = greenstickerPop


        HistInactive = histstickerPop - HistActive


        GreenInactive = greenstickerPop - GreenActive
        RedInactive = redstickerPop - RedActive

        TotalActive = GreenActive + RedActive + HistActive + PRedActive + PGreenActive
        TotalInactive = GreenInactive + RedInactive + HistInactive + PRedInactive + PGreenInactive



        'add active/inactive for the R/G OHMC and ATVs

        Me.Hide()
    End Sub


End Class
```