VIN Decoder

Executive Summary

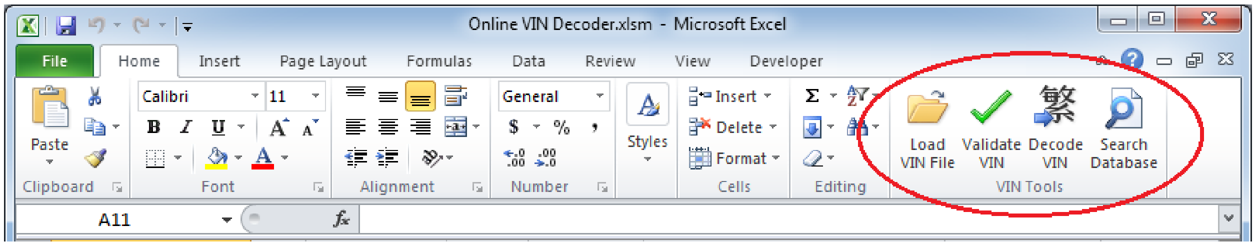
The Vehicle Identification Number (VIN) is a unique serial number that identifies every automobile manufactured from 1981 on to today. Prior to 1981, there was no accepted standard for these numbers, so different manufacturers used different formats. Since 1981, VINs consist of 17 characters which do not include the letters I (i), O (o), or Q (q) (to avoid confusion with numerals 1 and 0).The VIN serves as a serial number that contains information about the vehicle manufacturer, where the vehicle was manufactured, the manufacture year, and etc.

Although there are several websites that allow you to decode a VIN, none of the websites allow you to enter a list of VINs and have that list decoded. Dr. Snow of the Supply Chain department needed to decode over 250,000 VINs and it would take him a very long time to enter each one into a website, have it decode the VIN, copy the information, then sort it.

Since the information to write a VIN decoder is not easily available I wrote a program that reads a text file that contains multiple VINs, validates the VINs and uses a website (http://www.decodethis.com/) to decode the VIN. Once the VIN is decoded this information is downloaded and summarized. Once the list of VINs has been decoded the user can use a form to build a custom list of VINs and decoded information given the make, model, year, trim level, manufacture location, body style, engine type, and drive line.

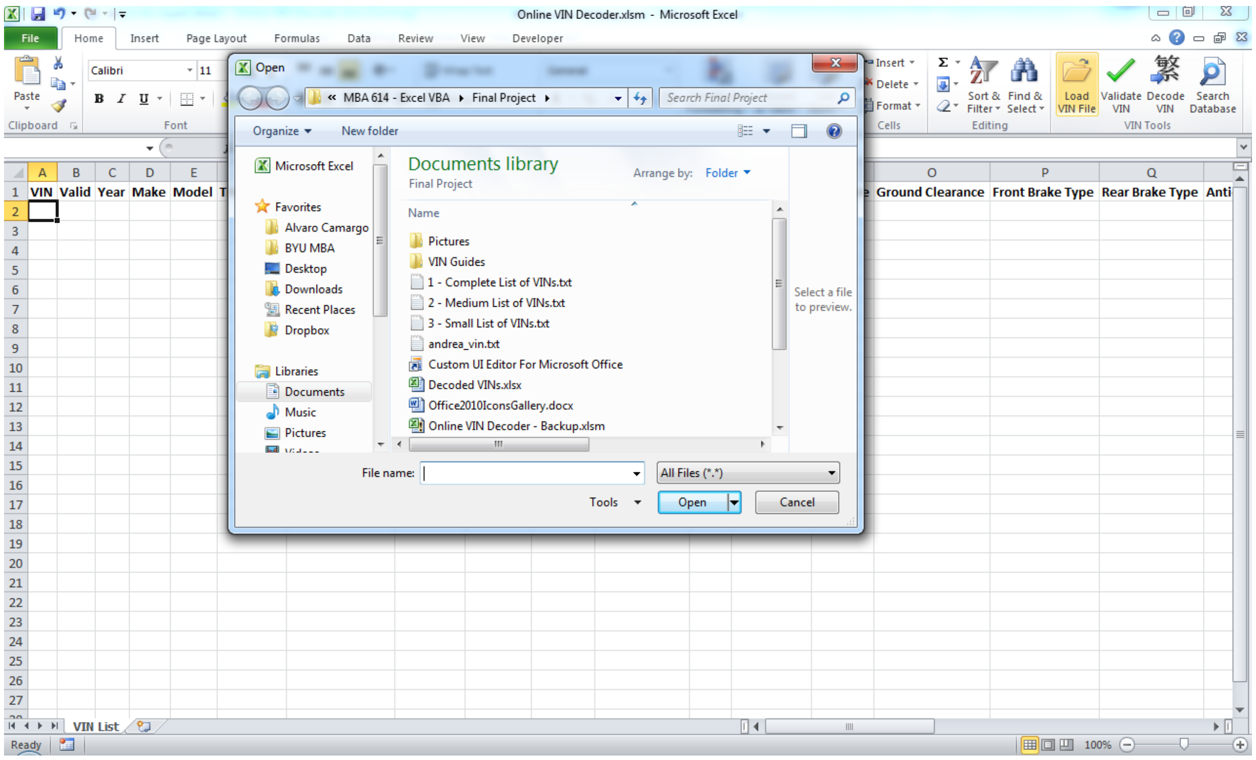
Implementation

In order to keep the workbook simple to use I divided the process of searching, downloading, and sorting the VIN data into 4 steps: Load VIN File, Validate VIN, Decode VIN, and Search Database. I started by creating a group on the Home tab of the Ribbon called VIN Tools and added 4 buttons called Load VIN File, Validate VIN, Decode VIN, and Search Database.



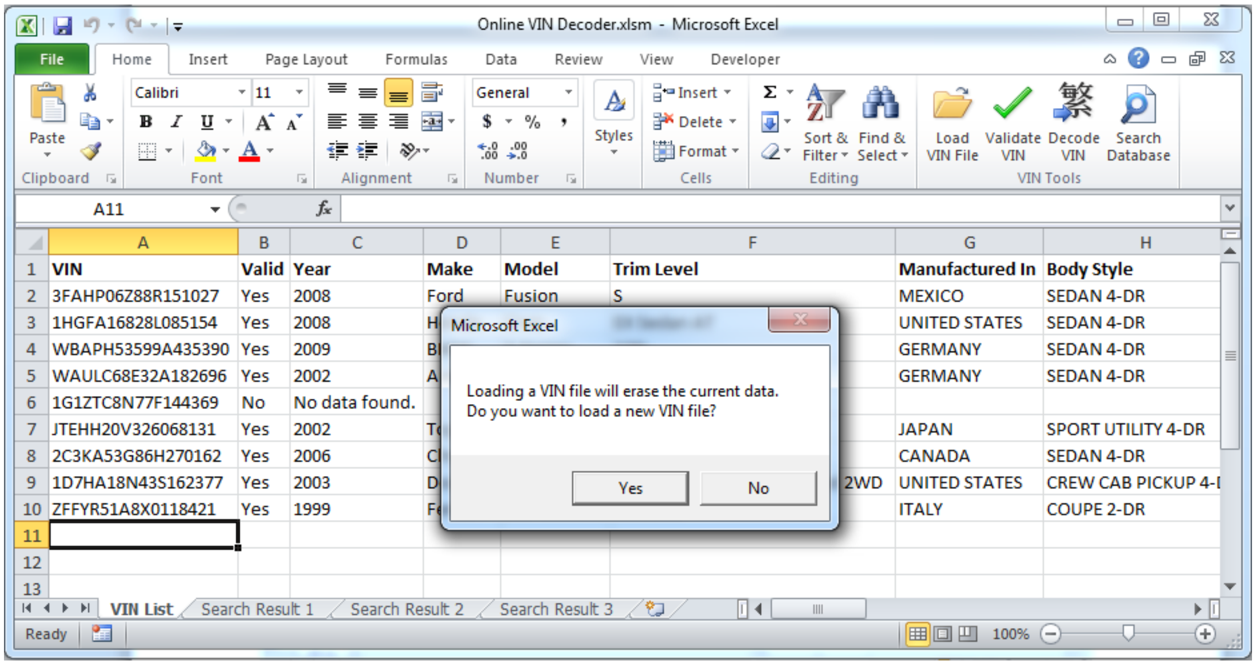
Load VIN File

The workbook needs to read am external file that contains the VINs and place them in column A starting at the second row (the first row is dedicated to the titles). In order to implement this functionality I used a File System Object which requires a reference to the Microsoft Scripting Runtime. The code calls the GetOpenFilename function of the Application class to launch the standard Windows Open window where the user can browse to and select the file they want to use.


Online VIN Decoder.xlsm - Microsoft Excel
Open
MBA 614 - Excel VBA ▸ Final Project ▸
Documents library
Final Project
Pictures
VIN Guides
1 - Complete List of VINs.txt
2 - Medium List of VINs.txt
3 - Small List of VINs.txt
andrea_vin.txt
Custom UI Editor For Microsoft Office
Decoded VINs.xlsx
Office2010IconsGallery.docx
Online VIN Decoder - Backup.xlsm
File name:
All Files (*.*)
Open
Cancel
Load VIN File
Validate VIN
Decode VIN
Search Database
VIN Tools
VIN List

The name and path of the file are saved in a variable and passed to the OpenTextFile method which allowed me to read each line of the file and place its content in a different row of column A.

Since every time a new file is loaded the previous data is deleted, when the “Load VIN File” button is clicked the code checks if there is any data in the sheet and warns the user that by loading a new file the current data will be erased.



	A	B	C	D	E	F	G	H
1	VIN	Valid	Year	Make	Model	Trim Level	Manufactured In	Body Style
2	3FAHP06Z88R151027	Yes	2008	Ford	Fusion	S	MEXICO	SEDAN 4-DR
3	1HGFA16828L085154	Yes	2008	H			UNITED STATES	SEDAN 4-DR
4	WBAPH53599A435390	Yes	2009	B			GERMANY	SEDAN 4-DR
5	WAULC68E32A182696	Yes	2002	A			GERMANY	SEDAN 4-DR
6	1G1ZTC8N77F144369	No	No data found.					
7	JTEHH20V326068131	Yes	2002	T			JAPAN	SPORT UTILITY 4-DR
8	2C3KA53G86H270162	Yes	2006	C			CANADA	SEDAN 4-DR
9	1D7HA18N43S162377	Yes	2003	D		2WD	UNITED STATES	CREW CAB PICKUP 4-I
10	ZFFYR51A8X0118421	Yes	1999	F			ITALY	COUPE 2-DR
11								
12								
13								

Microsoft Excel

Loading a VIN file will erase the current data.
Do you want to load a new VIN file?

Yes No

Validate VIN

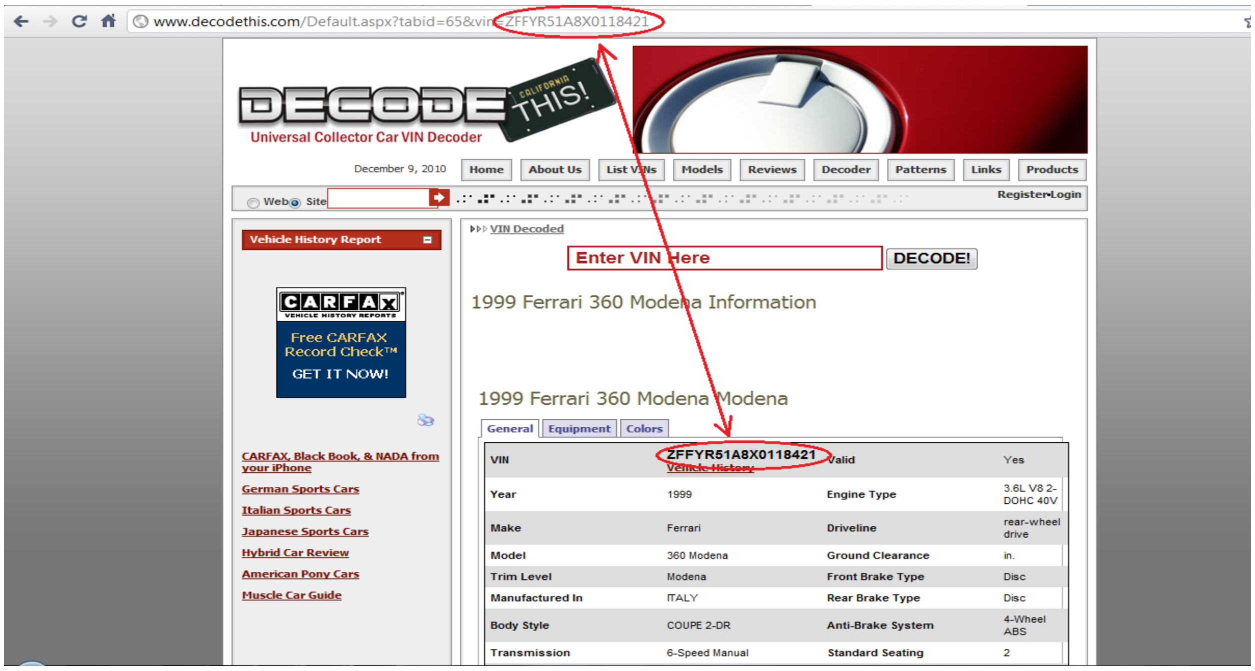
I started out the project with the intention of creating a self-contained excel file that could decode any VIN. After several hours searching the web I only found information to decode Ford vehicles. There is a lot of partial information for decoding VINs of the other vehicle manufacturers; however, since Dr. Snow needed to correctly decode VINs from several manufacturers I decided to use an online VIN decoder. The problem with using an online decoder is that it takes about 10 seconds to send the VIN to the online decoding tool, retrieve the VIN data, and place the data in the correct cells (the bulk of the time is spent waiting for the website to decode the VIN).

Since every VIN has a check digit (the 9th digit) that can be used to validate the VIN, I found out how the validation is done and created a subroutine (VINValidator) that does just that. Manually validating the VINs allowed me to speed up the decoding process by only submitting valid VINs to the online decoder.

Decode VIN

Clicking on the “Decode VIN” button will launch code that will first check if the VINs have already been validated. As described above, validating the VINs manually allows the code to only submit valid VINs to the online decoder thus speeding up the decoding process. If the VINs have been validated the code simply sends the valid VIN to the website. If the VINs have not been validated the code validates the VIN, then if the VIN is valid, sends it to the online decoder.

Choosing the VIN decoding website was also a though out process. There are several websites that decode VIN but the one I chose uses the VIN as part of the URL that decodes VINs.

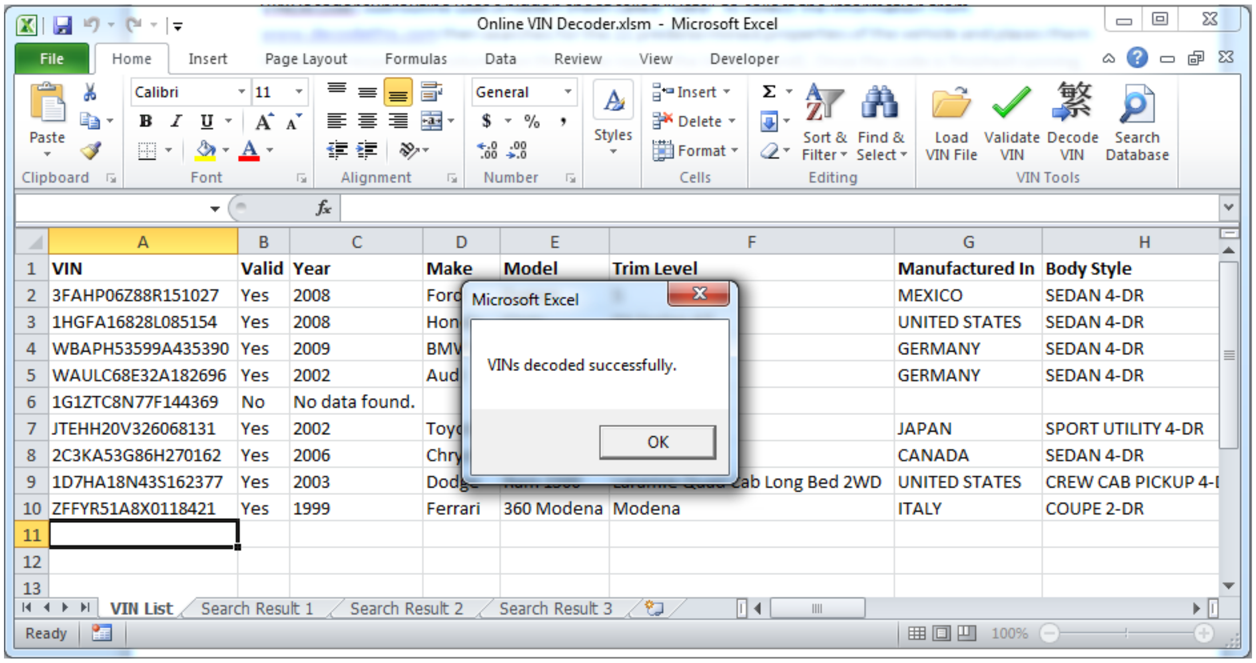


1999 Ferrari 360 Modena Information

1999 Ferrari 360 Modena Modena

VIN	ZFFYR51A8X0118421 Vehicle History	Valid	Yes
Year	1999	Engine Type	3.6L V8 2-DOHC 40V
Make	Ferrari	Driveline	rear-wheel drive
Model	360 Modena	Ground Clearance	in.
Trim Level	Modena	Front Brake Type	Disc
Manufactured In	ITALY	Rear Brake Type	Disc
Body Style	COUPE 2-DR	Anti-Brake System	4-Wheel ABS
Transmission	6-Speed Manual	Standard Seating	2

By choosing this website (http://www.decodethis.com/) I was able to avoid having to manipulate Internet Explorer manually and simply used the Excel function “Get External Data – From Web.” The VINDecoder subroutine uses a hidden sheet called “Data” to collect the information from www.decodethis.com then searches for the 22 predetermined properties of the vehicle and places them in their corresponding column (on the same row as the VIN is found). Once the code is finished running, a message indicates the completion to the user.



	A	B	C	D	E	F	G	H
1	VIN	Valid	Year	Make	Model	Trim Level	Manufactured In	Body Style
2	3FAHP06Z88R151027	Yes	2008	Ford			MEXICO	SEDAN 4-DR
3	1HGFA16828L085154	Yes	2008	Hon			UNITED STATES	SEDAN 4-DR
4	WBAPH53599A435390	Yes	2009	BMV			GERMANY	SEDAN 4-DR
5	WAULC68E32A182696	Yes	2002	Aud			GERMANY	SEDAN 4-DR
6	1G1ZTC8N77F144369	No	No data found.					
7	JTEHH20V326068131	Yes	2002	Toyo			JAPAN	SPORT UTILITY 4-DR
8	2C3KA53G86H270162	Yes	2006	Chry			CANADA	SEDAN 4-DR
9	1D7HA18N43S162377	Yes	2003	Dodge		ab Long Bed 2WD	UNITED STATES	CREW CAB PICKUP 4-I
10	ZFFYR51A8X0118421	Yes	1999	Ferrari	360 Modena	Modena	ITALY	COUPE 2-DR
11								
12								
13								

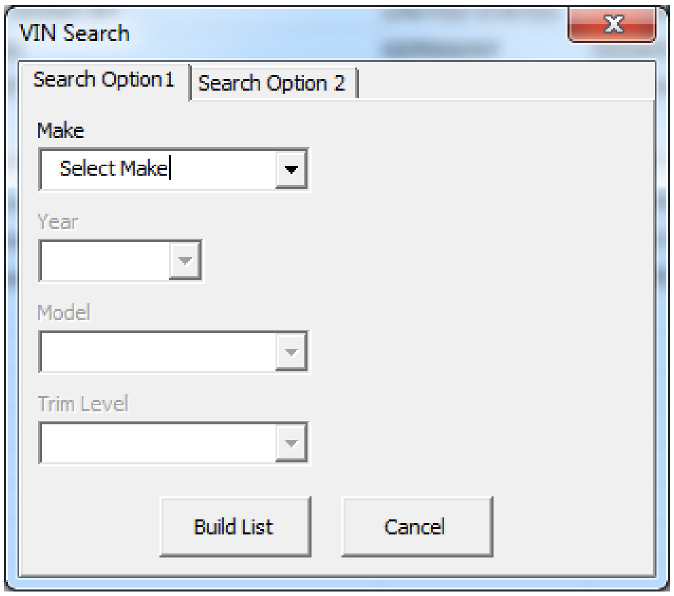
Microsoft Excel

VINs decoded successfully.

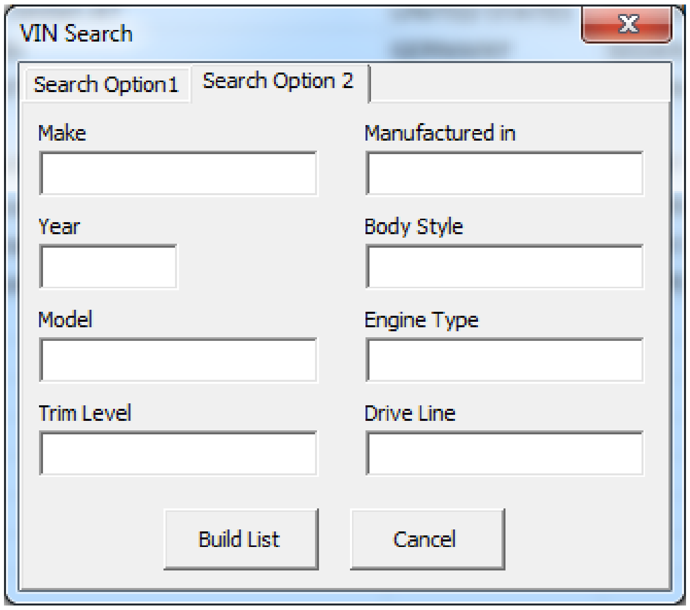
OK

Search Database

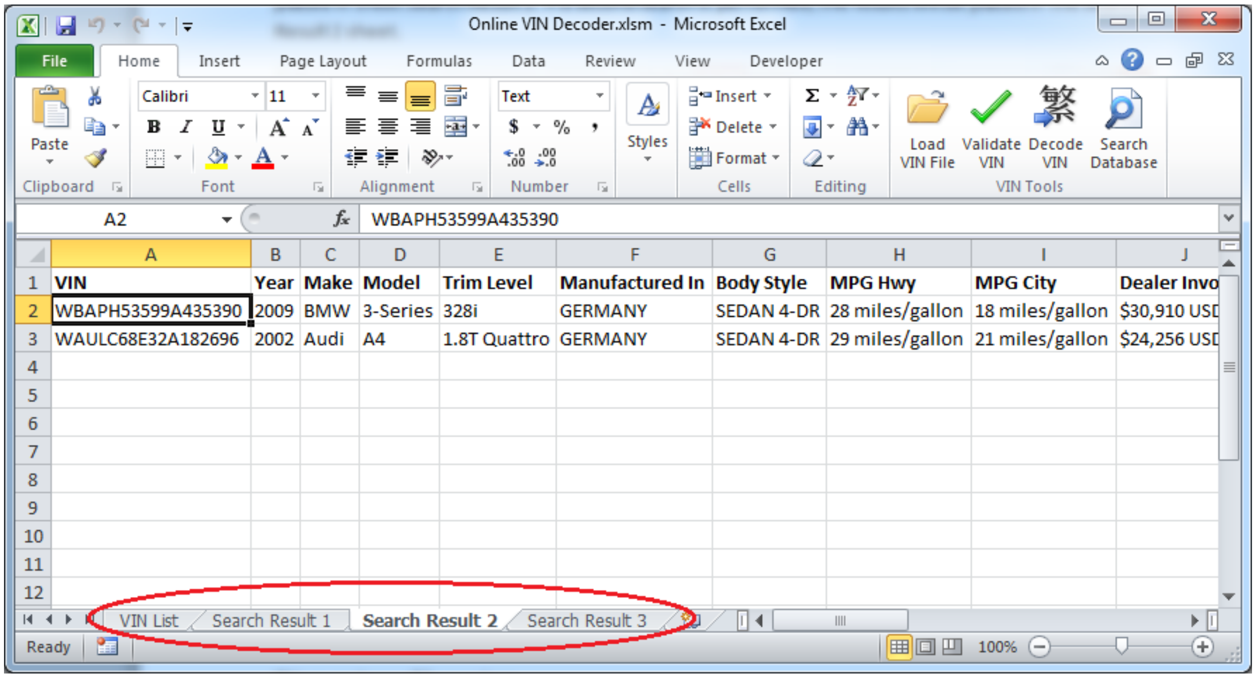
Once all VINs have been decoded the user can use a form to search for a VIN (or group of VINs) by using 8 different properties. The Form has two tabs. In the first tab (Search Option 1) the user uses dropdown boxes to select the search options based on 4 properties (Make, Year, Model, and Trim).



The user can enter a value for all properties of a combination of four properties and the code will fetch the data. In the second tab (Search Option 2) the user has 8 properties that she can select from to build her search. The code will search for VINs that fit all of the search criteria (for example, if Make and Engine are being searched then only VINs that meet the Make and Engine search terms will be selected).



The results of the search are placed in a new sheet called “Search Result #.” The first search will be placed in sheet Search Result 1. If a second search is performed, the results will be placed in the Search Result 2 sheet.



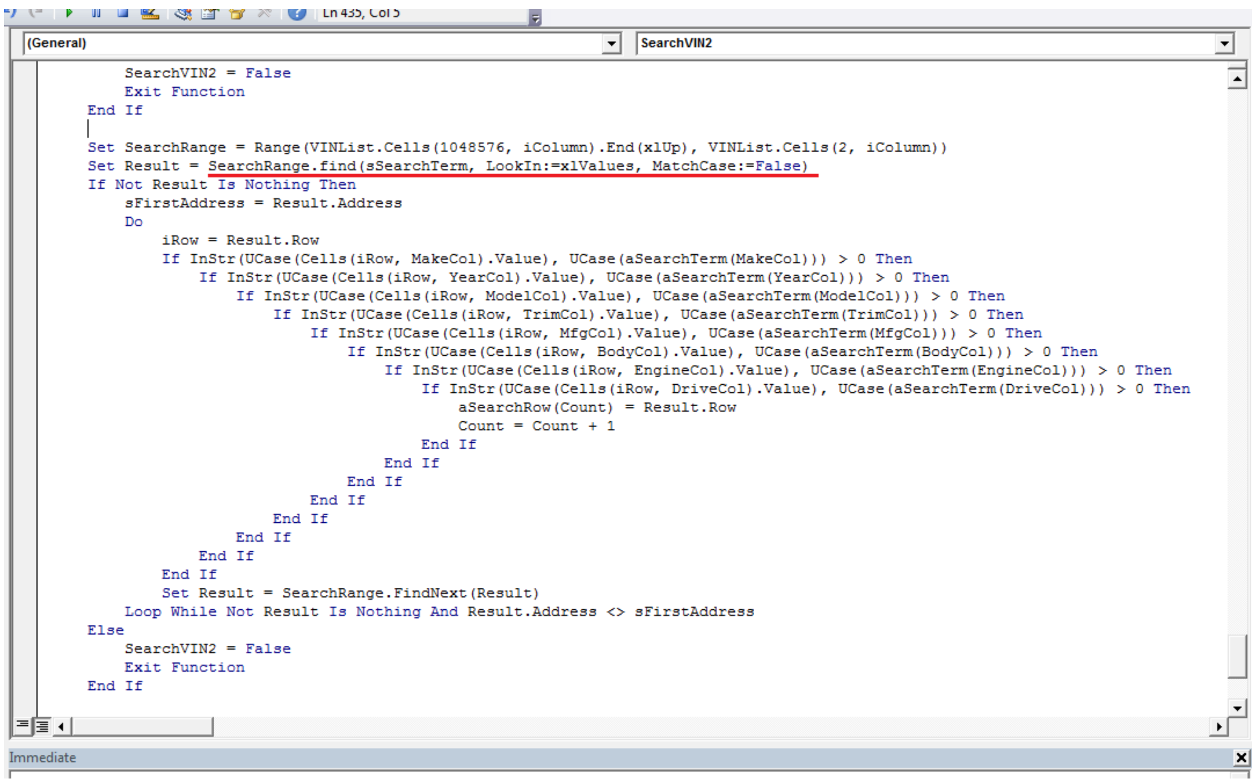
WBAPH53599A435390

	A	B	C	D	E	F	G	H	I	J
1	VIN	Year	Make	Model	Trim Level	Manufactured In	Body Style	MPG Hwy	MPG City	Dealer Invo
2	WBAPH53599A435390	2009	BMW	3-Series	328i	GERMANY	SEDAN 4-DR	28 miles/gallon	18 miles/gallon	$30,910 USD
3	WAULC68E32A182696	2002	Audi	A4	1.8T Quattro	GERMANY	SEDAN 4-DR	29 miles/gallon	21 miles/gallon	$24,256 USD
4										
5										
6										
7										
8										
9										
10										
11										
12										

Discussion of learning

I used many of the objects, methods, properties, forms, and coding techniques we learned in class during the semester to create this however, since this program will have to handle a large amount of data (over 250,000 VINs) I had to make sure the code was as effective as possible in order to conserve memory and time. One of the new practices I learned was to clear objects, variables, and arrays. By “collecting the garbage” I could prevent memory leaks.

Another thing I learned was how to effectively search for items in a sheet. In our USDADirectory assignment we used a form to find a combination of 3 search terms. In my implementation of the solution I created a subroutine that would loop through each row looking for the search data. Then I created 7 cases (one case for each combination of search field) each containing a different set of code to find the searched terms. This time around I used the Range.find method and several nested IF statements to make the search more effective.



```
            SearchVIN2 = False
            Exit Function
        End If
        
        Set SearchRange = Range(VINList.Cells(1048576, iColumn).End(xlUp), VINList.Cells(2, iColumn))
        Set Result = SearchRange.find(sSearchTerm, LookIn:=xlValues, MatchCase:=False)
        If Not Result Is Nothing Then
            sFirstAddress = Result.Address
            Do
                iRow = Result.Row
                If InStr(UCase(Cells(iRow, MakeCol).Value), UCase(aSearchTerm(MakeCol))) > 0 Then
                    If InStr(UCase(Cells(iRow, YearCol).Value), UCase(aSearchTerm(YearCol))) > 0 Then
                        If InStr(UCase(Cells(iRow, ModelCol).Value), UCase(aSearchTerm(ModelCol))) > 0 Then
                            If InStr(UCase(Cells(iRow, TrimCol).Value), UCase(aSearchTerm(TrimCol))) > 0 Then
                                If InStr(UCase(Cells(iRow, MfgCol).Value), UCase(aSearchTerm(MfgCol))) > 0 Then
                                    If InStr(UCase(Cells(iRow, BodyCol).Value), UCase(aSearchTerm(BodyCol))) > 0 Then
                                        If InStr(UCase(Cells(iRow, EngineCol).Value), UCase(aSearchTerm(EngineCol))) > 0 Then
                                            If InStr(UCase(Cells(iRow, DriveCol).Value), UCase(aSearchTerm(DriveCol))) > 0 Then
                                                aSearchRow(Count) = Result.Row
                                                Count = Count + 1
                                            End If
                                        End If
                                    End If
                                End If
                            End If
                        End If
                    End If
                End If
                Set Result = SearchRange.FindNext(Result)
            Loop While Not Result Is Nothing And Result.Address <> sFirstAddress
        Else
            SearchVIN2 = False
            Exit Function
        End If
```

Conclusion

We were expected to spend 40 hours working on this project and I spent just about that. Coming in to this project I thought I would be able to finish it quickly, but having to search for non-existing decoding instructions took up quite a bit of time. Another time consuming portion of the project was making sure the code was as effective as possible in order to avoid memory leaks and conserve time. Although I had previous experience working with Excel VBA before starting this class, this project has definitely been the largest Excel VBA project I've worked on and the one that I'm most proud of.