Suggestion	Rationale	Example
Keep all of your run specifications in the same directory as the import files associated with that RunSpec. This folder should be outside the folders installed by MOVES so that it will not be overwritten when installing a new version of MOVES.	MOVES will first look in the location of the RunSpec when browsing for import files, so keeping them all in one place keeps you from searching for the files that will be imported.	My Documents\MOVES\ [CountyYear]
Decide on the Scale and Calculation Type you wish to use before making selections on any other panels.	Changing the Scale or Calculation Type option after making choices in other panels may require you to re-enter data for those panels.	
Using the description panel is helpful when trying to discriminate between several runs. Think about filling out some sort of repeatable formula such as date, relevant important parameters, who is doing the run, etc. Note that MOVES preserves these descriptions in the movesrun output table for future reference.	Documentation of the RunSpec is very important, especially if many scenarios are being processed.	2010_01_01; John Doe; Washtenaw County; 2010; July
Every run used for any official purpose should be archived completely. All batch files, RunSpecs, MOVES code and configuration files, supplemental code copied into MOVES by batch files, default database, user input databases, etc. should be saved. File location can be added as the "Description" when the file is imported.	Months or years later, the run can be reproduced by anyone opening the archive.	

Suggestion	Rationale	Example
If they are not too large, output related runs should be saved to the same MySQL database	Combined with useful Run descriptions, you can keep all the data, and its description in a single place. Looking at the Moves Run table in that database can be very informative.	
Always "GROUP BY" MOVESRunID, PollutantID, and DayID when summing output results.	If a table includes output from multiple runs, it is very easy to mistakenly sum them together, thereby drastically increasing emissions. Emissions from different pollutants should never be summed. The types of day must be correctly weighted before they can be summed.	SELECT MOVESRunid, DayID, Pollutantid, sum(emissionquant) FROM movesoutput GROUP BY MovesRunid, DayID, Pollutantid;

Appendix F - Scenario 1: Estimating Changes in CO2 using the AVFT

Explanation of Scenario 1: Estimating Changes in CO2 using the AVFT

In this example run specification (AVFT_Example_1_runspec.mrs), the EPA's Motor Vehicle Emission Simulator (MOVES) was used to generate estimates of light-duty vehicle greenhouse gas (GHG) emissions and fuel use. The "policy scenario" discussed here consists of potential increases in the stringency of car and truck CO_2 standards, while the baseline scenario is similar to the MOVES defaults. To model the policy scenario, new MOVES inputs were created to simulate potential vehicle CO_2 limits.

Creating new MOVES inputs involved multiple steps. First, Alternative Vehicle Fuels and Technologies (AVFT) files were created to model the CO_2 limits. As explained below, the same AVFT file was also used to specify the fraction of the vehicle fleet in a given calendar year that is comprised of vehicles of a certain vehicle technology (e.g., conventional gasoline, conventional diesel, electric vehicle, etc.).

(A) AVFT files were used to define a new baseline diesel percentage which is higher than that shown in the MOVES 2010 default database.

(B) For simplicity, CO_2 limits were simulated by shifting a fraction of the gasoline and diesel vehicle fleets into the electric vehicle (EV) fleet; since EVs do not have tailpipe CO_2 emissions, shifting vehicles to the EV fleet has the same effect as reducing the tailpipe CO2 emissions from the overall fleet. It should be noted this approach is not predicting that EVs will be used to meet tighter CO_2 limits, it is simply the method used here to achieve the desired overall fleet CO_2 levels.

Use of the AVFT:

Incorporating potential CO_2 limits into the MOVES AVFT files was achieved by shifting a fraction of the gasoline and diesel fleets into the electric vehicle fleet. The basis of the MOVES CO_2 estimates is energy consumed and the carbon content of various fuels. By switching a certain portion of vehicles to a fuel with zero tailpipe emissions, we reduce the overall fleetwide tailpipe CO_2 emissions.

Practically, we can model new potential CO_2 limits in MOVES by modifying fuel consumption. Because tailpipe CO2 emissions are directly proportional to liquid fuel consumption, our discussion focuses on miles per gallon (MPG).

Table **F-1** shows the fuel economy assumptions for cars in the baseline and policy scenarios. These fuel economy values are unadjusted values (i.e. the type of values used in Corporate Average Fuel Economy (CAFE) standards), so they are higher than the actual onroad fuel economy. Real world conditions create a gap between on-road fuel economy (calculated by MOVES) and the CAFE values which were used to generate the percent reductions in fuel consumption.

Table F-1
MPG values for Baseline and Policy Scenarios

Model Year	Cars	
	baseline	*Policy Scenario*
2001	29.8	29.8
2002	29.8	29.8
2003	29.8	29.8
2004	29.8	29.8
2005	29.8	29.8
2006	29.8	29.8
2007	29.8	29.8
2008	29.6	29.6
2009	29.7	29.7
2010	29.7	29.7
2011	30.1	30.1
2012	30.4	30.4
2013	30.7	30.9
2014	30.7	32.2
2015	30.7	33.5
2016	30.7	34.8
2017	30.7	36.2
2018	30.7	37.6
2019	30.7	39.1
2020+	30.7	40.7

For simplicity, the baseline and policy scenario assumes equivalent diesel penetration, as shown in Table F-2. The penetration set in the current example differs from that used in the default MOVES database, and is set by the new AVFT.

Table F-2
Diesel market share (%)

Model year	Cars
2001 - 2010	1.0
2011	1.2
2012	1.3
2013	1.3
2014+	1.3

For a given model year, the percentage of vehicles of each fuel type was calculated based on the scenario fuel economy, and the diesel market share, as follows:

$$\text{Scenario } \%EV = \left[1 - \left(\frac{\frac{1}{ScenarioFuelEconomy}}{\frac{1}{DefaultFuelEconomy}} \right)\right],$$

For example, the percentage of EVs for cars in MY 2014, for the policy scenario, was:

$$\text{Scenario } \%EV = \left[1 - \left(\frac{\frac{1}{32.2}}{\frac{1}{30.7}} \right)\right] = 4.66\%,$$

The percentages of gasoline and diesel vehicles in the car and truck fleets were then calculated such that the percentages of diesels in the car and truck fleets, excluding the EVs, equaled the market share percentages specified in Table F-2, as follows:

$$\%diesel = (1 - \%EV) \times DieselMarketShare$$
$$\%gasoline = (1 - \%EV) \times (1 - DieselMarketShare)$$

For example, in the MY 2014 cars policy scenario, the gasoline and diesel percentages were:

%diesel	= (1-0.047) x .013 = 1.24%
%gasoline	= (1- 0.047) x (1-0.013) = 94.10%

Analysis of Results

The following instructions assume that your results are created in the default database name of CO2_test_database, and that there is only a single run residing in this database. For simplicity, the analysis focuses upon model year 2014.

1. You can check to see that the electric vehicles produce no CO2.

```
SELECT * FROM CO2_test_database.movesoutput m where yearid=2014 and
modelyearid=2014 and sourcetypeid=21 and pollutantid=90;
```

Look at the emissionquant column, and match it to the various fueltypes. Notice that for fuel type 9 emissions equal 0 for both starts and running, while fuel types 1 and 2 have emission output.

2. Go to the movesoutput table, and select pollutant id for CO_2 (90) along with the start and running processes (1 and 2). Sum the emission quantity (emissionquant) and record this value.

```
SELECT sum(emissionquant) FROM CO2_test_database.movesoutput m where yearid=2014
and modelyearid=2014 and sourcetypeid=21 and pollutantid=90 and fueltypeid in (1,2);
```

3. Look at the movesactivityoutput table and in the year 2014, sum the VMT from the cars with the modelyear 2014. Record this value.

```
SELECT sum(activity) FROM CO2_test_database.movesactivityoutput m where yearid=2014
and modelyearid=2014 and sourcetypeid=21;
```

By including the electric fueled mileage in the activity output, you can simulate a CO_2 standard for cars which is higher than the MOVES default. Dividing the emission output by the activity provides grams of CO2 per mile. This number can be compared against a similar run without an AVFT in order to compare the potential CO_2 improvements over a baseline scenario. Remember, the MOVES output is based on-road fuel consumption, which is higher than indicated by CAFE standards.

Appendix G - Scenario 2: MOVES Project Level Example

1. Definition of the MOVES Project

In this example MOVES Runspec, the EPA's Motor Vehicle Emission Simulator (MOVES) was used to generate emission inventory estimates for a hypothetical scenario where heavy-duty vehicle traffic is entering and departing a parking lot using a single roadway. Since this example is only for illustration of the MOVES software features, it was kept simple intentionally. It does <u>not</u> represent any specific real-world project.

The Project constraints are:

1. There are only two links (inbound and outbound from a parking lot)
2. The two links and the off-network link are independent of each other.
3. A single off-network link contains all of the parking, extended idle and vehicle start operations.
4. Only heavy-duty vehicles operate on the roadway and are present in the off-network area.
5. The project example only models nitrogen oxide (NOx).
6. One hour of operation was selected. If the user desired to model additional hours, then additional MOVES Projects need to be created.

Figure G1 Basic Schematic of the Project

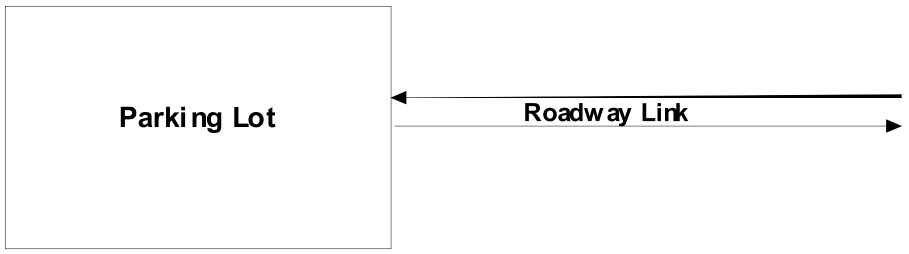


Figure G1 shows a basic schematic of the Project to be modeled. It consists of a parking lot and a two-way road link leading in and out of the parking lot. The specific modeling parameters are shown and discussed in Tables G1, G2 and G3 below.

Table G1
Summary of On-road Project Level Parameters

Location County	Washtenaw County, Michigan
Calendar Year	2009
Month	January
Time	11:00 PM to 11:59:59 PM (hour 24)
Weekday/Weekend	Weekday
Temperature	23.7 F
Humidity	74.5 %
Road type of links (link1 and link 2)	Rural Unrestricted Access – can represent any non freeway rural road
Roadway Link Length (both links)	0.56 miles
Link Traffic Volume – In (link 1)	100 vehicles per hour
Link Traffic Volume – Out (link 2)	10 vehicles per hour
Link Average Speed – In (link 1)	33.6 miles per hour
Link Average Speed – Out (link 2)	31.6 miles per hour
Number of Vehicle in Parking Lot (link 3)	200 vehicle on average

Only one county may be chosen for a given project level run. In this example, Washtenaw County, Michigan was chosen. Any of the 3,222 counties or a custom domain (defined by the user) may be chosen. Also, only one calendar year, one month and one hour may be chosen for a given project level run. If the user desires to model more than one such entity (i.e., all 24 hours), then they must perform additional and separate project level runs. It is suggested for such runs that the user employ MOVES batch mode features. A project level run may include either weekday activity, weekend activity or both. In this example only weekday was run. A temperature in degrees Fahrenheit (23.7 F), and relative humidity of (74.5 %) were also specified. These are the default MOVES values for this county, month and hour. However, any appropriate value may be entered for these parameters.

More than one pollutant / process combination may be modeled in a given project level run if the required inputs are entered. In MOVES, some pollutants are 'chained' to other pollutants. This means that they require the prerequisite calculation of another pollutant(s). If the pollutant to be modeled in the Project level is a 'chained' pollutant, then a complete set of Project level inputs must be entered for all of the required prerequisite pollutants.

Currently, MOVES Project level cannot model evaporative emission processes. However, this capability will be added to future model upgrades.

For each roadway link the user must specify a MOVES road type which best represents it. In this example, the rural unrestricted access road type was chosen to represent both links. Any of the four roadtypes may be chosen to represent each Project link. The user must also specify a link length in miles for each roadway link they wish to model. In this case because the example is modeling the same roadway in and out of the parking lot, both links are 0.56 miles in length.

The traffic volume must also be specified for each link. This is the total average traffic flow from all vehicle types on the link during the hour period. In this simple example, all of the vehicles are heavy-duty trucks, but in general any or all of the MOVES vehicle source types may be included at the same time in a project level run.

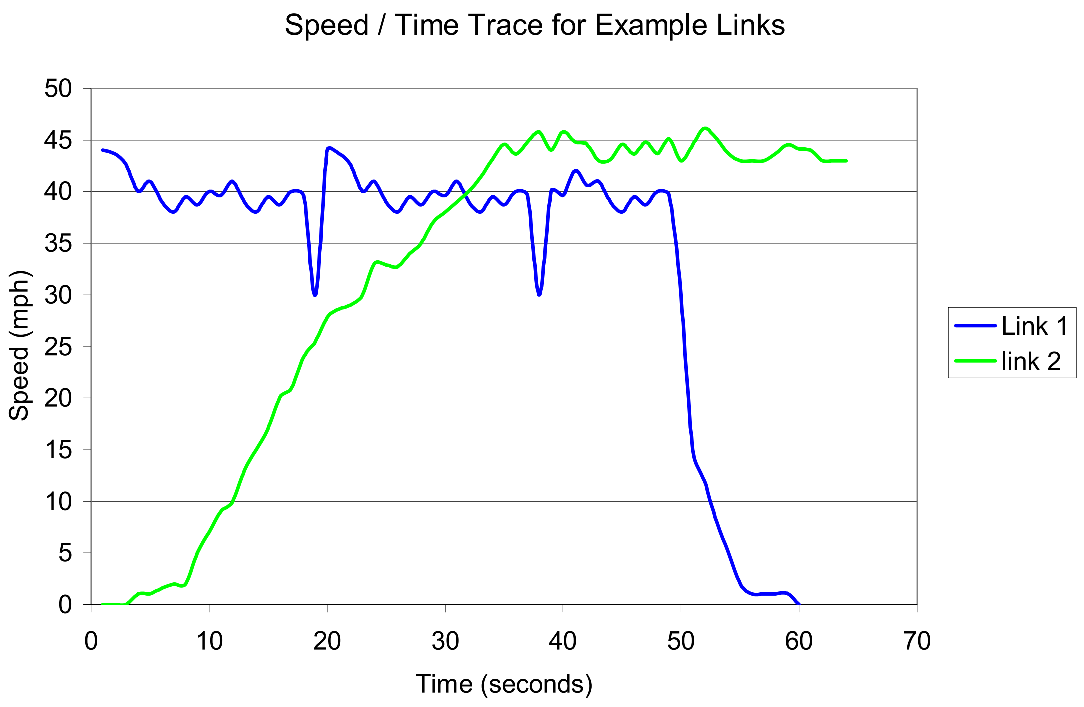
The average speed on each link must also be specified. In this example where drive schedule inputs are used to represent activity of all of the roadway links, the average speed inputs in the Links spreadsheet / input GUI tab are used only for minor internal calculations in the project level model. However, they should match the overall average speed(s) of the individual drive cycles as submitted in the LinkDriveSchedule tab. It is the responsibility of the user to insure that they match – no automatic checks are done.

The average road grade (in percent grade units) may also be specified for each link. This input represents the overall average grade of the entire link not one specific section of it. It is used only if a drive schedule input is not provided (i.e., it will NOT be used in this example). However, in this example a value of zero was entered as a placeholder.

In this example, MOVES will use second-by-second driving schedules to model vehicle operation. If drive schedules are not provided, MOVES uses the average speed and average grade inputs plus default MOVES driving cycles to model the driving behavior. When drive schedules are entered, different emission results may be produced even if the average speed of the driving schedules match the average speed input entered in the Links spreadsheet tab. See Appendix G Section 5.0 for an example.

Figure G2 shows the two driving schedules of this example project in graphical form. Link 1 shown in blue, has driving starting at around 44 mph and decreasing to zero mph as the vehicles enter the parking lot. Link 2 shown in green has driving starting at zero mph and accelerating to a 44 mph cruise. The variable nature of the speed – time curves are indicative of average operation of a number of vehicles / drivers on a road. The driving schedule data and the chart are contained in the input workbook "ProjectExample1_input.xls" in the worksheets "LinkDriveSchedule and LinkDriveSchedule Example Chart."

Figure G2



The distribution of the traffic by MOVES source type is an additional input. It is entered as the SourceTypeHourFraction in the LinkSourceType worksheet, and the LinkSourceType input tab. In this example, because only heavy-duty long-haul trucks are to be modeled, all of the values for SourceTypeHourFraction are set to unity.

For this example, the off-network parameters are shown in Table G2. These four parameters include a vehicle population of 200 vehicles in the parking lot on average during the hour period. This input is an average value over the one hour time period, because some vehicles may have been in the lot at all times while others entered or exiting during the hour period. In this example, the start fraction is set to three percent (0.03). This is the fraction of the average vehicle population which were started during the hour. If all of the vehicles in the lot on average are started, then this value is unity (1.0). It may also be greater than unity if large numbers vehicles are repeatedly started during the period.

The extended idle and parked vehicle fraction parameters were set to 0.90 and 0.09 respectively, for this example. The 90 percent input for extended idle reflects the fact that 90 percent of the total vehicle – hours (only one hour by definition) in the parking lot were spent in extended idle mode (vehicles are parked in a lot with their engines idling at higher than curb idle speeds). The 9 percent input for parking reflects the fact that 9 percent of the total vehicle – hours in the parking lot were spent in park mode (vehicle is parked and the engine is off). The sum of extended idling and parking cannot be greater than unity. *Currently, the parked vehicle input is not active in any of the calculations, but a valid entry must be provided anyway.*

Table G2
Summary of Off-Network Project Level Parameters

Average Vehicle Population	200 vehicles in the parking lot on average
Start Fraction	0.03
Extended Idle Fraction	0.90
Parked Vehicle Fraction	0.09

The operating mode distribution for the parking lot link must be entered. This is a distribution by AVERAGE vehicle 'soak' time (i.e., the time since a vehicle was last started). For example, Table G3 below shows that 5 percent of the vehicles in the parking lot have not been started for more than 720 minutes or 12 hours. Extended Idle operating model fraction is always set to unity (1.00) when it is present because it is the only operating mode of its type. This input should not be confused with the off network parameter called Extended Idle Fraction with a value of 0.90 that is shown in Table G2.

Table G3
Operating Mode Distribution Parameters for Start and Extended Idle Processes

OpmodeID Code	Operating Mode Description	Operating Mode Fraction
101	Soak Time < 6 minutes	0.00
102	6 minutes <= Soak Time < 30 minutes	0.05
103	30 minutes <= Soak Time < 60 minutes	0.30
104	60 minutes <= Soak Time < 90 minutes	0.10
105	90 minutes <= Soak Time < 120 minutes	0.50
106	120 minutes <= Soak Time < 360 minutes	0.00
107	360 minutes <= Soak Time < 720 minutes	0.00
108	720 minutes <= Soak Time	0.05
200	Extended Idle Operating Mode	1.00

Table G4 shows the source type age distribution for the vehicles in the project. In this example, there is only one source type (heavy-duty long haul trucks) present. Thus, only one age distribution is required. Additional age distributions would be required if additional source types were present. The age distribution runs from age zero (brand new) to 30 years. All ages greater than 30 years are included in the age 30 group. The distribution must sum to unity within a source type. Note that the project level model does not allow separate age distributions for different fuel types if the same source type is selected. For example, the same age distribution would be used for both gasoline and diesel long haul trucks if both were present in the project.

Table G4
Source Type Age Distribution

Source Type	ageID	ageFraction
62	0	0.2
62	1	0.15
62	2	0.1
62	3	0.1
62	4	0.1
62	5	0.07
62	6	0.05
62	7	0.05
62	8	0.05
62	9	0.02
62	10	0.02
62	11	0.01
62	12	0.01
62	13	0.01
62	14	0.01
62	15	0.01
62	16	0.01
62	17	0.01
62	18	0.01
62	19	0.01
62	20	0
62	21	0
62	22	0
62	23	0
62	24	0
62	25	0
62	26	0
62	27	0
62	28	0
62	29	0
62	30	0

2. Example Data Creation and Input

A MOVES Runspec **(called ProjectExample1.mrs)** was created to model this example MOVES Project level analysis. For this example, the user should load the Runspec - *Project Example1.mrs*.

The user should note that a Project level run MUST contain only ONE

County
Year
Month
Hour

The example Runspec was further simplified to include only one sourcetype, two roadtypes, one weekday/weekend combination and three pollutant / process combinations. The runspec is provided as an example and the user should use it to become more familiar with this example.

The Project level importer is accessed by either selecting "Project Domain Manager" from the "PreProcessing" menu item at the top of the GUI, or pressing the "Enter/Edit Data" button on the "Geographic Bounds" panel. After one of these selections is made, the Project level importer screen will open.

Caution! In general the user should complete ALL of the runspec entries (i.e., Scale, Time, Geographic Bounds, etc) BEFORE accessing the Project Domain Manager.

The first input step is to create the Project Level database where the imported data is stored. This is a MySQL database and it is named "ProjectExample1_input" in this example.

The user loads the data into the Project Level database (i.e.,ProjectExample1_input) for each input tab by browsing for the individual file (use the Browse button), and once found, pressing the "Import" button to import the data into MOVES. A message diagnostic of "Import Complete" will be issued, and the GUI tab should turn from red to green if the data import was successful. This process is repeated for each of the tabs until no more red tabs are present. Various diagnostic messages are provided if the import was not successful. The data should be "Browsed" and "Imported" from the Excel workbook *ProjectExample1_input.xls*.

The inputs are in the spreadsheets.

1. DriveScheduleSecondLink
2. OffNetworkLink
3. ZoneMonthHour
4. IM / Reflash
5. LinkSourceTypeHour
6. SourcetypeAgeDistribution
7. FuelSupply
8. Link
9. OpModeDistribution

After the data loading process is complete the user should press the *Done* button to save the data in the database. This will exit the Project Level data manager. As a final step, the user may be required to go to the Geographic Panel tab and choose the new Project Level database **("ProjectExample1_input")** from a list of databases. If the new database is not visible, the user should press the *Refresh* button.

Both the Excel input file **ProjectExample1.xls** and the MOVES project level input database **ProjectExample1_input** are provided in the MOVES package as an assistance to the user. If the user "Browses" and "Imports" their data from the spreadsheet **ProjectExample1.xls,** the user's final database should be the same as **ProjectExample1_input**.

In this example, all of the inputs for the Project Level were read from a single Excel workbook. This was done to make the process easier for the first time user, and to facilitate and simple "Browse" and "Import" data entry. However, if the user is starting from scratch with their own project, it is recommended that they use the *Create Template* button or the *Export Default* button for each of the Project Level input tabs. This will create a set of Excel workbook templates or default data spreadsheets (with a set of individual worksheets - one for each input tab). The user should populate or modify these templates or data tables with the actual data. The spreadsheets within each template will provide necessary descriptions of the data fields and moves codes (i.e., countyID codes, fuelformulationID codes, roadtypeID codes, hourDayID codes and sourcetypeID codes). The MOVES project level importer will show a list of all of the individual Excel worksheets from an Excel workbook. The user must pick the 'active' worksheet (usually the leftmost worksheet in the workbook or the first spreadsheet entry in the MOVES importer) as an input

After using the Project level importer, the user should also notice that the DriveScheduleSecondLink and the opmodeDistribution table need not exist together for all of the roadtype links. In this example, the DriveScheduleSecondLink table contains an average speed-time 'trace' for both roadway links. It is used by MOVES to internally create an opmodeDistribution for each roadway link. A separate opmodeDistribution for the road links is not required to be supplied, but could have been supplied in-lieu of the DriveScheduleSecondLink data input. The opmodeDistribution is always required for the start operation parameters (table 3 contains the inputs). It contains the opmodeDistribution for the NOx emission starts. These values are used to differentiate start soak times.

Alternatively, neither a DriveScheduleSecondLink or the OpModeDistribution table need be provided. The user may simply specify an average speed for each link in the "Links" table. If this option is utilized, MOVES selects two default driving schedules and uses the average speed input to interpolate between the two cycles to create a new default cycle for that average speed. (See Appendix G Section 4 for more details).

3. MOVES Results for the Example

The results for this simple MOVES Project Level simulation are reported in the MySQL database "**ProjectLevel_Output**". They are summarized below.

Table G5
MOVES Project Level Example Simulation Results

LinkID	Link Description	NOx emissions (grams)
1	Inbound Road	793.31
2	Outbound Road	120.97
3	Parking Lot (extended idle)	9768.33
3	Parking Lot (start)	4.17

The emissions are reported as an 'inventory' for the project and are broken down by linkID (roadway and off-network – the user must specify that the results to be reported by roadway and process in the Runspec). In this example, the results are reported in units of grams of NOx emissions. The time domain for the Project Level is always one hour. The user should also remember that these results are **average** results over the one hour time domain and the geographic link domain. One reason project level reports 'average' results is because the model's calculation methodology assumes (for simplicity) independence of the individual links in the project, and does not dynamically calculate traffic flows between links or residence times in off network (the user must do this step **before** the data is entered into the MOVE project level simulator.

The large difference in NOx emission inventories between two physically similar links (link 1 and link 2) is because of the large difference in traffic volumes. The traffic volume for link 1 is an order of magnitude greater than the traffic volume for link 2. The relatively small amount of NOx emissions from the start process is the result of a very low start fraction input (i.e., 3 percent). The relatively large amount of emissions from the extended idle process reflects a fairly high fraction of extended idle operation (i.e., 90 percent). In comparison, the running operation of the inbound road contains half as many vehicles (100 vehicles versus 200 in extended idle), and the running operation on average lasts only about 2 minutes per vehicle versus the entire hour for 90 percent of the vehicles.

4. MOVES Project Example #2

This example (the inputs are in Excel Spreadsheet ProjectExample2_input.xls) has virtually all of the same inputs as Project Example #1 discussed in Sections 1 through 4. As a result none of the data and explanations will be repeated. The only difference between the two examples is the ACTIVE use of the Average Speed input in this example instead of the DriveSchedule inputs. The purpose of this brief example is to show that different results for roadway links are obtained from the two different methods of modeling vehicle operation.

In this example, the average speed input of 33.6 mph is imported and used for Link 1 and 31.6 mph is used for Link 2. These inputs are in the Links tab. The average speed / average grade inputs become ACTIVE in MOVES when the specific drive schedule inputs shown in Figure 1 are **NOT** imported / used by the Driveschedule Importer GUI. If the drive schedule data (or operating mode distribution data for roadway links) are imported, they automatically over-ride the average speed inputs in the Links tab.

Instead of user supplied drive schedules being used to represent vehicle operation, built-in default drive schedules are used when average speed inputs are entered. The two input types (drive schedule and average speed) are not necessarily equivalent. The difference can be quantified by comparing the results in Table G6 with those of Table G5.

Table G6
MOVES Project Level Example Simulation Results

LinkID	Link Description	NOx emissions (grams)
1	Inbound Road	731.77
2	Outbound Road	79.45
3	Parking Lot (extended idle)	9768.33
3	Parking Lot (start)	4.17

5. MOVES Project Example #3

This example briefly presents and discussed several real world ramp cycles developed under contractor for EPA. These are provided because the Project Level does not contain default data for modeling freeway ramps. Entering an average speed / grade in the Project Level ***Links*** importer will not model a freeway ramp. The only way a user can model a freeway ramp using the Project Level is to enter a ramp drive schedule in the ***DriveSchedule*** importer or a roadtype operating mode distribution using the OpmodeDistribution input tab.

The attached Excel spreadsheet ***Ramp_Driving_Cycles.xls*** contain several real world drive schedules for various ramp configurations. They are based on a 2004 Sacramento, California driving study. These schedules do not model any actual real world project, but serve as realistic examples of ramp operation. They also function as a starting point for the user to develop actual cycles that precisely fit the needs of their specific project. They can be used in MOVES Project (or modified for use) by defining a new ramp LinkID (i.e., Ramp1, Ramp2, etc) and populating the DriveSchedule importer with a speed-grade / time trace based on ramp driving operation.

The ***Summary*** table in the ***Ramp_Driving_Cycles.xls*** spreadsheet lists the ramp configuration type, the grade change toggle and the ramp metering designation. The table includes only "on-ramps". No data are available for "off-ramps". Ramps may be configured in several ways with different operating behavior occurring on each. The most prominent types are the "Diamond" configuration or the "Loop" or "Cloverloop" configuration. The "Diamond" configuration typically has higher speed and acceleration operation than the "Loop" configuration. A ramp may also be "metered" by a traffic light at the end of the ramp that controls access to the freeway. The final ramp configuration is the "Transition" ramp. It occurs when one freeway transitions to another freeway. The individual cycles are also denoted by the general grade change of the ramp. Three possibilities are provided for the "Diamond" and "Loop" configurations.

Appendix H - Scenario 3: County Data Manager Example and Basic MySQL Queries for Analyzing Output

1. Explanation of Scenario 3

The County Data Manager is expected to be used extensively by users, particularly by nonattainment or maintenance areas when conducting SIP or conformity analyses. This tool provides the user the ability to import and edit local data, rather than using MOVES defaults. This appendix will also go through the MOVESOutput table in the MySQL Query Browser to give users some basic queries that can be used for their own analyses.

In this scenario, the County Data Manager is used to construct a RunSpec using area-specific data. The area being modeled will be a custom domain. While there are some differences between using the custom domain option and selecting an individual county, the basic operation of the County Data Manager is the same. Where differences exist, this scenario will describe how a custom domain and a single county would be treated differently.

2. Constructing the RunSpec

The custom domain in this scenario will be named “Example City” and will be an ozone nonattainment area, so the selections and use of the County Data Manager will reflect that assumption. On the **Pollutants and Processes** panel, all processes for Volatile Organic Carbons (VOCs) and Oxides of Nitrogen (NO_X) are selected because these pollutants are associated with the formation of ground-level ozone. If VOCs are selected, several other pollutants and processes must be selected as well because they are needed when calculating VOC emissions. Therefore, this run contains a fair number of pollutant-processes and will take a fair amount of time to run (about 16 minutes on a computer with a dual-core processor), but the variety of the output will be useful later when analyzing the output in the MySQL Query Browser. If the user simply wants practice with the process of using the County Data Manager and importing files, it is recommended that the user select only NO_X processes on the **Pollutants and Processes** panel as evaporative emissions of hydrocarbons take the most time in MOVES.

The RunSpec should be filled out in its entirety before the County Data Manager is used. In this example, selections were limited to reduce the run-time of the RunSpec, but some variation is included to emphasize how output can be post-processed in MySQL.

Open the file name “examplecity_2013_july.mrs” (located in the USER Guide Example Files\City Example\City Example Files folder) to view the RunSpec and to create an output database using the General output panel (e.g. examplecity_2013_july_out). Then select the RunSpec **Geographic Bounds** panel and notice that Custom Domains require the user provide additional information about the area. Users can change any of these fields (except County ID because the input files have countyID=99001), but should be aware that doing so will result in different emission results. Next, click the “Enter/Edit Data” button to open the County Data Manager window.

3. Using the County Data Manager

With the County Data Manager open, name and create a user-input database (e.g. examplecity_2013_july_in).

Caution! If any changes are made to the RunSpec after the database has been created, the database may no longer be correct for the revised RunSpec. This is because some "core" tables are populated based on the selections in the RunSpec at the time of the creation of the database and these core tables can only be changed directly in MySQL (i.e., there are no tabs to edit these core tables). Users should take care in if changing the RunSpec while using an existing database and should always double-check the tab headers to ensure all the necessary data has been provided after making a change to the RunSpec.

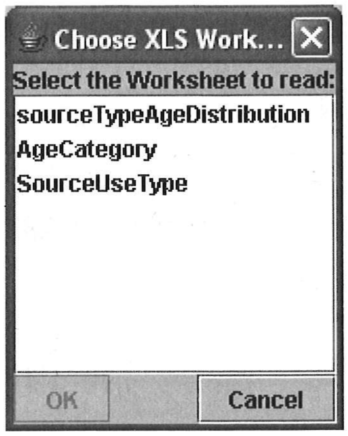
Next, notice that the Fuel Formulation, Ramp Fraction, Zone Road Activity, and I/M Programs tabs all start with green checks for different reasons. For Fuel Formulation, as long as default fuel formulations are referenced in the Fuel Supply tab, then no data has to be imported; however, if a new fuel formulation is created or if the properties of an existing fuel formulation are changed, then the new fuel formulations should be imported. For Ramp Fraction, the default fraction of 8% of VHT is applied if no data is imported on this tab, but users can import new values if desired. The Zone Road Activity tab erroneously shows a green check, but data MUST be imported for MOVES to calculate results properly; a value of 1 should be entered for the SHOAllocFactor for each road type so that all of the VMT input by the user is assigned to the custom domain. Finally, the I/M Programs tab starts with a green check because there are no default programs in the custom domain that have to be accounted for. Of these 4 tabs, data will be imported for Zone Road Activity and I/M Programs.

For all other tabs, files containing the area-specific data must be imported. The importing process is the same for all the tabs, so it is explained generally in the next paragraph. The Excel files containing the example area-specific data are included as part of the MOVES installation pack and have the following naming convention:

examplecity_2013_july_[TABNAME]

Users can open the files before importing them to view how the data is formatted and can refer to the supporting worksheets for clarification on the numerical codes or abbreviations. Notice that for some files, wildcards are used to greatly reduce the number of rows that have to be entered (e.g. Average Speed Distribution - 64 with wildcards vs. 11,520 without).

Importing the files is straightforward - simply click the "Browse" button, find where the file is saved, and select the appropriate file for the tab (i.e. if working in the "Age Distribution" tab, select the file "examplecity_2013_july_agedistribution.xls"). After the file has been selected, a small pop-up window appears:



Select the correct worksheet (in this example, it will generally be the first worksheet listed and will usually have a name similar to the tab name) and click "OK." For the Vehicle Type VMT tab, there are four worksheets within the file "examplecity_2013_july_vehicletypevmt.xls" that should be imported: HPMSVTypeYear, monthVMTFraction, dayVMTFraction, and hourVMTFraction. The names of the worksheets correspond to the names of the database tables for which data is being provided.

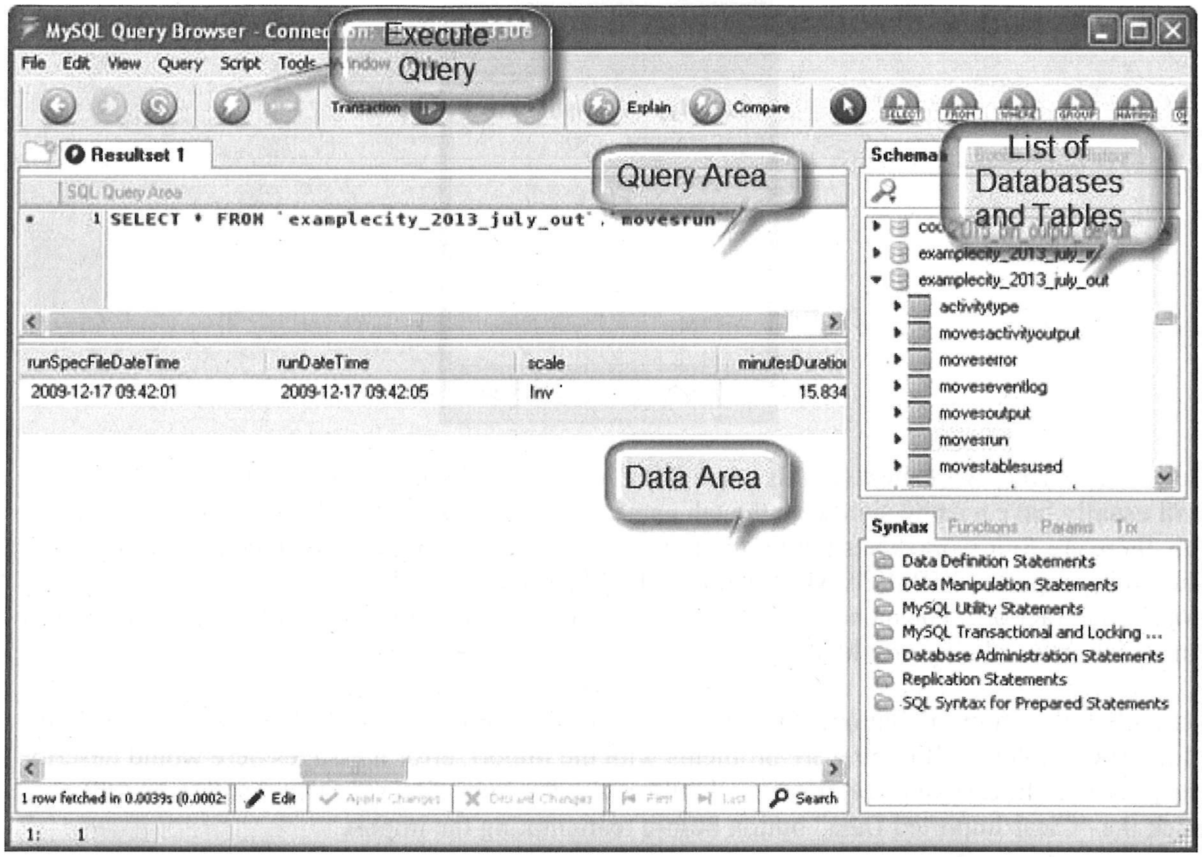
Next click the "Import" button. The "Messages" box should say which table was imported and "Import Complete." If there were problems with the import, error messages here would describe the problem. In case of error, the user should revise the input file, save the changes, and then click the "Clear Imported Data" button before reattempting the import.

The County Domain Manager can be used to input data for a custom domain (as shown in this example) or a specific county. Currently, the only difference is that the custom domain option includes a "Zone Road Activity" tab. This tab exists to allow the Source Hours Operating (SHO) to be allocated to different zones within a custom domain; however, this capability is not fully implemented in MOVES2010 and values of 1 should be entered for all road types within the zone.

Once all the files have been imported, the database tables have been populated. Click "Done" in the County Data Manager to return to the **Geographic Bounds** panel. Make sure the user-input database is selected in the **Domain Input Database** section of the **Geographic Bounds** panel. The RunSpec and county-input database have both been completed, so the RunSpec can now be executed by selecting **Execute** from the **Action** drop-down menu.

4. Analyzing the Results in the MySQL Query Browser

First, a brief summary of the major pieces of the MySQL Query Browser.



```
SELECT * FROM `examplecity_2013_july_out`.`movesrun`
```

```
runSpecFileDateTime     runDateTime            scale    minutesDuratio
2009-12-17 09:42:01     2009-12-17 09:42:05    Inv             15.834
```

```
1 row fetched in 0.0039s (0.0002
```

The top left box is the Query Area, bottom left is the Data Area, and top right is the List of Databases/Tables. The tables within a database can be revealed by clicking the black triangle just to the left of the database name. There are queries below that have the following syntax:

SELECT * FROM `examplecity_2013_july_out`.`[table]`;

These queries can be auto-generated by dragging the table name from the right-hand List of Tables and dropping it in the Query Area or Data Area. If the table name is dropped in the Query Area, the text for the query is generated, but the query it self is not executed; so the user must either click the Execute button (lightning bolt) or press CTRL + Enter. However, if the table name is dropped in the Data Area, the query will be generated in the Query Area and auto-executed with results immediately appearing in the Data Area.

Now that some of the basics have been covered, the results from the example can be analyzed. Listed below are MySQL Queries and some discussion of the results they generate. Users should be able to copy and paste the queries into the Query Area of MySQL as long as the output database name is "examplecity_2013_july_out"; if the user provided a different output database name, that name should be used wherever "examplecity_2013_july_out" appears.

1) SELECT * FROM `examplecity_2013_july_out`.`moveserror`;

Users can look at the MOVESError log to make sure the run executed normally. There should be no results from this query.

2) SELECT * FROM `examplecity_2013_july_out`.`movesrun`;

The MOVESRun table gives some basic information about the RunSpec, the description, the date and length of time, etc. If there are multiple runs in the database, there will be multiple MOVESRunID entries here.

3) SELECT * FROM `examplecity_2013_july_out`.`movesactivityoutput`;

The MOVESActivityOutput table contains all the activity results from the RunSpec and should have 1446 rows (there is a bar below the data area that tells you how many rows a query returns). In the RunSpec, the distance traveled and population options were selected. Both of these values were inputs, so let's make sure the input values were carried through to the output correctly. Population is easier, so we'll look at that first.

4)
```
SELECT sourcetypeID, SUM(activity) AS population
FROM `examplecity_2013_july_out`.`movesactivityoutput`
WHERE activitytypeID=6
GROUP BY sourcetypeid
ORDER BY sourcetypeid;
```

You'll notice the query has gotten more complicated, but it can be easily explained. The first line is the SELECT statement, which states the fields from the table that will be displayed. Since population is not time dependent and we're only looking at one county, the only important field is the sourcetypeID itself; however, a sum of the activity is required because output was differentiated by fuel type, but that is not of interest right now. The SUM(activity) will be renamed in the resulting table "AS population". The second line is the FROM statement, which just identifies the database and table that should be used. The third line is the WHERE clause, which specifies that only activitytypeID=6 should be used when querying the table (since we're just looking at population). The fourth line is the GROUP BY clause, which is used when a sum is calculated. Since the clause says, "GROUP BY sourcetypeid", all data of different source types will be kept separated, but any data within the same source type will be summed (so the different populations for each fuel type will be added together). The fifth row is the ORDER BY clause, which simply states the order in which the resulting information will appear.

The resulting table from this query should have just 3 rows and the populations should be (after rounding) 500,000; 300,000; and 50,000 for source types 21, 31 and 32, respectively. These values match the populations that were imported.

5)
```
SELECT dayid,sourcetypeid, SUM(activity)*dayID*31/7 AS PortionofMonthVMT
FROM `examplecity_2013_july_out`.`movesactivityoutput`
WHERE sourcetypeid=21 AND activitytypeid=1
GROUP BY dayid
ORDER BY dayid;
```

This statement is the beginning to calculate the VMT for passenger cars (sourcetypeid=21), but doing so is not a straight-forward process and that fact can be seen in the SUM statement above (notice that the GROUP BY clause only has dayid because the WHERE clause identifies that only 1 sourcetypeid will be queried, so grouping by sourcetypeid is not necessary, but would not affect the query if it were included). The temporal output for the RunSpec is hour, so that means the VMT output was originally for one hour of one of the types of days; therefore, the VMT should be summed within the dayID to get the total for that type of day. Then, this query adds two additional steps: 1) by multiplying by the dayid, the product is the VMT on that type of day in a week; and 2) by multiplying by 31/7 (the number of days in July divided by the number of days in the days in a week), which is the number of weeks in July, this second product gives the VMT on that type of day in the entire month.

The resulting table should have just 2 rows as can be seen in the table below. An additional column has been added using an Excel spreadsheet to sum the VMT from the two types of days.

dayid	sourcetypeid	PortionofMonthVMT	MonthVMT
2	21	1.21E+08	
5	21	3.44E+08	4.65E+08

The monthVMTFraction table assigned a value of 1 to July, so the VMT that was input was essentially a monthly VMT for July and we can see that the MonthVMT is equal to the value that was input for HPMSVTypeID of 20 (examplecity_2013_july_vehicletypevmt.xls, worksheet "HPMSVTypeYear"). There are ways to calculate the MonthVMT entirely within MySQL, but that involves either creating tables or using subqueries, which is beyond the scope of this example. Therefore, once users get to this point, use of an external spreadsheet tool is recommended, unless the user is familiar with more complex MySQL queries.

More detail can be acquired using the query above as a template (e.g., adding "roadtypeID" to the SELECT, GROUP BY, and ORDER BY clauses to see the VMT on each road type to make sure the road type distribution was applied correctly; calculating VMT for HPMSVTypeID=30). Users are encouraged to experiment with the query above to improve their MySQL skills and better understand how to compare inputs with outputs to make sure the results are logical.

```
6)      SELECT * FROM `examplecity_2013_july_out`.`movesoutput`;
```

This table should contain 23432 rows and gives all the emissions output with a significant amount of detail since output is differentiated by fuel type, emission process, road type, and source type. Therefore, using MySQL to condense the results holds great potential for simplifying the post-processing.

```
7) SELECT     monthid, pollutantid, SUM(emissionquant) AS grams
        FROM `examplecity_2013_july_out`.`movesoutput`
        WHERE dayid=5 AND pollutantid IN (3,87)
        GROUP BY dayid, pollutantid
        ORDER BY pollutantid;
```

This query will greatly simplify the original output table from 23,000 rows to 2, but a major reason for this is that not all the data is being reported.

monthid	pollutantid	grams
7	3	2.11E+07
7	87	1.21E+07

Some assumptions were made to reduce the reported output that may be similar to those users will make in their analyses, such as only a single weekday is being looked at and the only pollutants of interest are NO_X (3) and VOCs (87). Notice that in the WHERE clause that a new command was used, “pollutantid in (3,87)”; the IN command is a more concise way of saying “pollutantid=3 AND pollutantid=87” and can be used with any field that has multiple entries.

Once again, users are encouraged to vary this query to see the results for different conditions (e.g. choose other pollutants; see emission results from each source type or on the different road types; etc.).

Appendix I – Stage II Refueling Control Programs

Stage II refueling emission control programs are intended to reduce HC and associated air toxics emissions by reducing the amount of gasoline vapor that escapes to the atmosphere during refueling. The amount of reduction depends on whether the vehicle has an onboard recovery system and the level of uncontrolled emissions. The uncontrolled emissions are calculated from inputs such as fuel RVP, vehicle fuel economy, and temperature parameters. Stage II programs are run by state, local or tribal governments and the effects of these programs will vary depending on the number of locations and the sales volume of stations equipped with Stage II recovery systems and their average state of repair. See the MOVES Software Design Reference Manual for more information about how MOVES calculates emissions from refueling.

The MOVES database contains information about all of the existing Stage II programs by county based on the parameters used for the 2005 National Emisson Inventory (NEI). For the initial release of MOVES, the effects of Stage II programs can only be altered by users by manually editing the tables used by MOVES to obtain the Stage II control information. Future versions of MOVES are expected to include graphical user interface (GUI) tools to assist in altering Stage II program effects for individual counties. Until these tools are available, this appendix is intended to assist users in making changes to the default parameters related to Stage II program effects.

Stage II refueling emission control programs can only affect refueling losses that occur during refueling. Onboard Refueling Vapor Recovery (ORVR) systems on modern vehicles are designed to minimize the refueling losses without Stage II controls. These reductions are already accounted for by MOVES, so that the additional control of Stage II will only affect the remaining refueling losses from these ORVR vehicles.

County Year Table

The MOVES database contains a table named CountyYear which contains information about each county for every calendar year. This table contains two fields that determine the Stage II program effects:

o refuelingVaporProgramAdjustment
o refuelingSpillProgramAdjustment

The refuelingVaporProgramAdjustment field is a number between zero and one (1.0) which indicates the fraction that is the reduction in full refueling displacement vapor losses that result from the Stage II recovery program that county in that calendar year. The refuelingSpillProgramAdjustment field is a number between zero and one (1.0) which indicates the fraction that is the reduction in full refueling fuel spillage losses that result from the Stage II recovery program that county in that calendar year. A value of zero would indicate that the program had no effect and a value of one would indicate that all vehicle refueling emissions had been eliminated.

Updating Refueling Adjustments Using MySQL Query Browser

The simplest way to change an existing Stage II program effect is to use the MySQL Query Browser application to open the appropriate table and manually change the existing values, but NEVER change values in a default MOVES database provided by EPA. MOVES is designed to easily provide replacement values for default values using MySQL tables via the **Manage Input Data Sets** panel on the MOVES graphical user interface (GUI). Using the **Manage Input Data Sets** panel, type the name you wish to use for the database which will hold your Stage II estimates into the Database drop-down box. Now click on the "Create Database" button and your will create a database to hold your information. The name will have to meet the criteria for database names (no spaces, no special characters) and we suggest it include text to help you identify its contents and purpose (for example, StageII_Input).

Once you have an input database, you can open the MySQL Query Browser to work with the tables. If the Query Browser was already open, you may need to right-click in the Schemata panel and refresh the list of database. Once you find your database, double-click on the name and that will show you all of the tables it contains. The database will contain empty versions of every table used by MOVES. The table that contains the Stage II program adjustments is called CountyYear and contains only four fields:

- countyID
- yearID
- refuelingVaporProgramAdjustment
- refuelingSpillProgramAdjustment

Using the Query Browser Edit function (at the bottom of the result window) you can add or alter the contents of the table. Be sure to execute the Apply Changes button to save any alterations.

Now return to the Manage Input Data Sets panel and select your database from the pull down menu. You can add additional description text and then click the add button. When you include this database in your run specification, MOVES will use your values in preference to any default values in the MOVES database.

The other empty tables may be deleted from your database. Leaving them will not harm your run specification (since the tables are empty), but they can be confusing. To delete empty tables, go to the Query Browser, double click on the database to show the tables and right click on the table you wish to delete. Choose "drop table" from the menu and the table will be removed from that database. Multiple tables can be selected by using the CTRL or Shift buttons.

Appendix J – MOVES "Decoder"

MOVES "Decoder"

SourceType

sourcetypeid	sourcetypename
11	Motorcycle
21	Passenger Car
31	Passenger Truck
32	Light Commercial Truck
41	Intercity Bus
42	Transit Bus
43	School Bus
51	Refuse Truck
52	Single Unit Short-haul Truck
53	Single Unit Long-haul Truck
54	Motor Home
61	Combination Short-haul Truck
62	Combination Long-haul Truck

Process

processid	processName
1	Running Exhaust
2	Start Exhaust
9	Brakewear
10	Tirewear
11	Evap Permeation
12	Evap Fuel Vapor Venting
13	Evap Fuel Leaks
15	Crankcase Running Exhaust
16	Crankcase Start Exhaust
17	Crankcase Extended Idle Exhaust
18	Refueling Displacement Vapor Loss
19	Refueling Spillage Loss
90	Extended Idle Exhaust

Pollutant

pollutantid	pollutantname
1	Total Gaseous Hydrocarbons
2	Carbon Monoxide (CO)
3	Oxides of Nitrogen
5	Methane (CH4)
6	Nitrous Oxide (N2O)
20	Benzene
21	Ethanol
22	MTBE
23	Naphthalene
24	1,3-Butadiene
25	Formaldehyde
26	Acetaldehyde
27	Acrolein
30	Ammonia (NH3)
31	Sulfur Dioxide (SO2)
32	Nitrogen Oxide
33	Nitrogen Dioxide
79	Non-Methane Hydrocarbons
80	Non-Methane Organic Gases
86	Total Organic Gases
87	Volatile Organic Compounds
90	Atmospheric CO2
91	Total Energy Consumption
92	Petroleum Energy Consumption
93	Fossil Fuel Energy Consumption
98	CO2 Equivalent
100	Primary Exhaust PM10 - Total
101	Primary PM10 - Organic Carbon
102	Primary PM10 - Elemental Carbon
105	Primary PM10 - Sulfate Particulate
106	Primary PM10 - Brakewear Particulate
107	Primary PM10 - Tirewear Particulate
110	Primary Exhaust PM2.5 - Total
111	Primary PM2.5 - Organic Carbon
112	Primary PM2.5 - Elemental Carbon
115	Primary PM2.5 - Sulfate Particulate
116	Primary PM2.5 - Brakewear Particulate
117	Primary PM2.5 - Tirewear Particulate

Day

dayID	dayName
2	Weekend
5	Weekdays

Road Type

roadtypeid	roaddesc
1	Off-Network
2	Rural Restricted Access
3	Rural Unrestricted Access
4	Urban Restricted Access
5	Urban Unrestricted Access

FuelType

fuelTypeID	fuelTypeDesc
1	Gasoline
2	Diesel Fuel
3	Compressed Natural Gas
9	Electricity

Activity

activityTypeID	activityType	activityTypeDesc
1	distance	Distance traveled
2	sourcehours	Source Hours
3	extidle	Extended Idle Hours
4	sho	Source Hours Operating
5	shp	Source Hours Parked
6	population	Population
7	starts	Starts

SCCV Type

SCCVtypeID	PART5SCCV typeDesc	MOBILE6SCCVtypeDesc
1	LDGV	1, 'LDGV', 'Light Duty Gasoline Vehicles (LDGV)'
2	LDGT1	Light Duty Gasoline Trucks 1 & 2
3	LDGT2	Light Duty Gasoline Trucks 3 and 4
4	HDGV	Heavy Duty Gasoline Vehicles 2B thru 8B and Gasoline Buses
5	MC	Motorcycles (MC)
6	LDDV	Light Duty Diesel Vehicles (LDDV)
7	LDDT	Light Duty Diesel Trucks 1 thru 4 (LDDT)
8	2BHDDV	Heavy Duty Diesel Vehicles (HDDV) Class 2B
9	LHDDV	Heavy Duty Diesel Vehicles (HDDV) Class 3, 4, and 5
10	MHDDV	Heavy Duty Diesel Vehicles (HDDV) Class 6 and 7
11	HHDDV	Heavy Duty Diesel Vehicles (HDDV) Class 8A and 8B
12	BUSES	Heavy Duty Diesel Buses (School and Transit)