United States
Environmental Protection
Agency

Air and Radiation

EPA420-P-04-017
November 2004

MOVES2004 Software Design Reference Manual

Draft

EPA420-P-04-017
November 2004

MOVES2004 Software Design Reference Manual

Draft

Assessment and Standards Division
Office of Transportation and Air Quality
U.S. Environmental Protection Agency

Authors:

Mitch Cumberworth
John Koupal
Megan Beardsley
Harvey Michaels

Additional Software Designers:

Wes Faler, Cimulus, Inc.
David Brzezinski

A Note about the Capitalization and Naming Conventions Used in this Document

Because object orientation is central to the design of the MOVES software, this document often refers to classes and objects. The class names used in MOVES are often formed from several English words which are run together without spaces, e.g. “EmissionCalculator”, “RunSpecEditor”, etc. MOVES follows the widely-used Java naming convention of capitalizing each word in these run-together names, and this convention is also used in this document.

Because databases are also central to MOVES, this document includes many references to databases, tables, and fields. It is our intention that database and table names follow the same naming convention as classes in MOVES. The names of fields within these tables also use this scheme, except that they begin with a lower case letter (unless they begin with an acronym). E.g. there is a “MOVESDefault” database which contains an “EmissionRate” table and this table contains a field named “meanBaseRate”. We’ve applied the same naming convention to file and directory names as well because they are somewhat analogous to tables and databases.

Acronyms, such as “VMT” (from “vehicle miles traveled”) or “AC” (from “air conditioning”) are capitalized in all contexts, even when they begin a field name. Thus there is a field named “ACPenetrationFraction”. A table of acronyms used in this document is contained in Appendix III.

There are undoubtedly instances where this document fails to fully comply with these conventions. In some contexts it seems natural to use ordinary English, e.g. “emission rate”, interchangeably with a class name, e.g. “EmissionRate”, and we have not attempted to be highly rigorous in this regard. The Windows operating system is not case sensitive and MySQL is also rather forgiving in this respect. We hope our readers will be as well.

Table of Contents

1. Introduction

MOVES2004 is EPA's initial release of the MOtor Vehicle Emission Simulator. This initial release can be used to estimate national inventories and projections at the county-level for energy consumption, N_2O, and CH_4 from highway vehicles. It also includes an interface with an updated version of Argonne National Laboratory's model for Greenhouse gases, Regulated Emissions, and Energy uses in Transporation (GREET) to include life-cycle (i.e., well-to-pump) effects in estimates of energy consumption and emissions. Future versions of the model are planned to estimate pollutants from additional mobile sources such as aircraft, locomotives, and commercial marine activity, estimate non-highway mobile source emissions, estimate criteria pollutant emissions, and operate at smaller scales.

For a brief overview of MOVES2004 and its documentation the reader is urged to consult *A Roadmap to MOVES2004* which explains the intended role of each MOVES2004 document, including this one.

A precursor to this design manual is a report published in October 2002 entitled *Draft Design and Implementation Plan for MOVES*, available on the MOVES website (www.epa.gov/otaq/ngm.htm). This draft plan includes extensive background on the impetus for MOVES, an analysis of the "use cases" MOVES is attempting to address, and the conceptual design for the model. The reader is encouraged to consult this draft design plan if additional background is desired. The design presented in this document is based on that one. The draft plan underwent formal peer review and public stakeholder review; the comments resulting from this process are summarized in Appendix I.

This Software Design and Reference Manual has several purposes: One is to provide an idea of what is involved in configuring, installing, and using MOVES2004. This document is not specifically tailored to the beginning user or to getting started quickly using the model, but is intended to answer questions about the model software such as "What software licensing terms apply to MOVES2004?" or "Can my computer run MOVES2004?" or "How might I set up multiple computers to run this model?" Chapter 2 identifies the software and database components which make up MOVES2004. Chapter 3 covers the hardware and system software required to run each component.

Chapter 4 covers configuration of the MOVES software, which can be run on a single computer or a network of computers. Chapter 5 covers MOVES software licensing, and Chapter 6 provides an overview of the MOVES installation process. Another document, the *MOVES2004 Installation Guide* describes installation in step-by-step detail for the beginning user, and is intended to be used when actually performing installation. This Software Design and Reference Manual is also not intended to function a “User Guide” for MOVES2004. Information as to how to begin using MOVES2004, including the functionality of its Graphical User Interface, is contained in the *MOVES2004 User Guide.*

A second role of this document is to explain the design of the MOVES software including its databases. Chapter 7 provides a “processing overview” and Chapter 8 diagrams and discusses the flow of data and control between the components of MOVES. Chapter 9 discusses its functional design concepts and Chapters 11 and 12 explain its databases.

A third purpose, related to the second, is to document the software’s functionality: its inputs, the calculations it performs, and its outputs. Almost half of this document consists of its Chapter 10 which contains functional specifications for each software component of MOVES, including the calculations performed. Chapter 11 documents the MOVES2004 input and working database and Chapter 12 documents its output database.

Finally this document contains comments received and EPA’s responses to them. Appendix I summarizes comments received on the predecessor design document, the *Draft Design and Implementation Plan for MOVES*, dated October, 2002 and EPA responses to them. Appendix II contains the peer review comments received on an earlier draft of this document and EPA’s reponses, a number of which include improvements made to this version.

Other documents supplement this manual. Those most closely related to the software are listed below. Once MOVES has been installed, on-line help screens are also available to assist with more detailed operation of the MOVES Graphical User Interface (GUI).

The *MOVES Installation Guide*, already mentioned above, is included on the MOVES installation CD.

The *MOVES2004 User Guide*, already mentioned above, is included on the MOVES installation CD.

More detailed documentation on the MOVES database is included within the database itself.

Technical documentation explaining the default data sources and methods used to estimate fleet, activity, and emission data underlying MOVES2004 is contained in technical reports separately downloadable from the EPA web site.

Documentation produced by Argonne Laboratories covering the GREET model is contained in the GREET directory within the model itself.

2. MOVES2004 Software Components

MOVES2004 is written in Java™ and the MySQL relational database management system, a product of MySQL AB. Its principal user inputs and outputs, and several of its internal working storage locations, are MySQL databases. A "default" input database, covering 3222 counties of the United States and which supports model runs for calendar years 1999 - 2050 is included with the model. MOVES2004 interfaces with a version of the GREET model produced by Argonne Laboratories, which is included on the MOVES2004 installation CD.

MOVES2004 has a "master – worker" program architecture which enables multiple computers to work together on a single model run. A single computer can still be used to execute MOVES2004 runs by installing both the master and worker components on the same computer.

Looking at this architecture in greater detail, the MOVES2004 software application consists of eight components, each described briefly here.

MOVES2004 Graphical User Interface (GUI) and Master Program: This is a Java program which manages the overall execution of a model run. Its MOVES GUI (also sometimes referred to as the run specification editor) may be used to create, save, load, and modify a run specification or RunSpec, and to initiate and monitor the status of a model run. A basic command line interface may be employed (in lieu of the GUI) by users (or by other computer programs) to execute the model without interacting with the MOVES GUI. If MOVES is installed on a computer network several model runs may be executed concurrently by different copies of the MOVES 2004 Master Program.

MOVES2004 Worker Program: This is also a Java program. At least one executing copy of this program is needed to complete a MOVES run. It may execute on the same computer as the MOVES Master Program, or on other computer(s) having access to the SharedWork directory.

Default input database, normally named "MOVESDefault": this MySQL database must reside on the same computer as the MOVES Master Program. A version of this database is included in the MOVES2004 Installation Package.

SharedWork: This is a file directory or “folder” which is accessible to all executing copies of the MOVES Master program and to the MOVES Worker program. It is not a MySQL database but simply a file directory or “folder” provided by the file services of a software operating system.

Optional user input databases: These MySQL databases are normally located on the same computer as the MOVES Master Program. They may contain any of the same tables that are in the MOVESDefault database and are used to add or replace records as desired by the user.

The MOVESExecution database: This MySQL database is created by the MOVES2004 Master Program. It is used for temporary working storage and does not interact directly with the user. It must be on the same computer as the master program.

MOVES output databases: MySQL databases are named by the user and produced by runs of MOVES2004. While normally located on the same computer as the MOVES Master program, they may be located on any MySQL server accessible to it.

MOVESWorker database: This MySQL database is used as working storage by the MOVES2004 Worker Program. It does not interact directly with the user. It is on the same computer as the MOVES Worker program.

3. Computer Hardware and System Software Requirements

The MOVES2004 application software components require a computer hardware and software "platform" upon which to operate. The hardware platform can consist of a single computer system or a network of computers.

Computer(s) used to run either of the MOVES application programs must have at least 128MB of RAM, (256MB or more is recommended). Execution run time performance is a constraint with MOVES so high speed processor(s), at least 1GHz and preferably faster, are highly recommended. The MOVESDefault database distributed with MOVES requires approximately 125MB of disk storage. MOVES Worker and Output databases are also often voluminous, so several gigabytes of disk space should be available on all machines used to run either MOVES program. Serious users of MOVES2004 will want to use late-model, high-performance microcomputer systems.

3.1. Details on JAVA Platform Requirements

The MOVES GUI/Master and the MOVES Worker are Java programs and for operation require a Java RunTime Environment (also sometimes referred to as the "Java Virtual Machine"). The Java Software Development Kit (SDK) includes the Run Time Environment. The initial release of MOVES2004 uses version 1.4.2 of the Java SDK, produced by SUN Microsystems Inc. The MOVES Installation Package includes this Java version, suitable for installation on WINDOWS NT, WINDOWS 2000, and WINDOWS XP systems. MOVES2004 does not operate successfully on versions of WINDOWS that predate WINDOWS 98, and many WINDOWS 98 systems will not have sufficient processor speed, memory, and disk capacity to run MOVES2004 well. Users should not attempt to operate either MOVES2004 program with other versions. While Java is available for other software operating systems, such as LINUX, UNIX, etc. and porting MOVES to such software operating systems should not be difficult, EPA has not tested such configurations and is not prepared to support them. Sun's main web site for information related to Java is http://java.sun.com/ .

Several extensions to Java are also required by MOVES2004 and are included in its installation package. These include JUnit and JFCUnit, which facilitate software testing, and JavaHelp used to construct the on-line help facility in MOVES2004. The ANT software build utility is also included in the MOVES2004 installation package.

3.2. Details on MySQL Platform Requirements

The MySQL database management software has client-server architecture. The MOVES GUI/Master and MOVES Worker programs function as MySQL clients and require access to MySQL server(s). Since both programs require a MySQL database to be located on the same computer (MOVESExecution and MOVESWorker), all computers that run either the MOVES GUI Master program or the MOVES Worker program must also operate a MySQL server. Additional computers operating MySQL servers can also be utilized; e.g., for MOVES Output databases.

The initial release of MOVES2004 uses MySQL version 3.23.49. The MOVES Installation Package includes an installation package for this MySQL version, suitable for installation on WINDOWS NT, WINDOWS 2000 and WINDOWS XP systems. While this version of MySQL is now somewhat dated, users should not attempt to operate either MOVES2004 program with other versions. While MySQL is available for other software operating systems, such as LINUX, UNIX, etc. and porting MOVES to such software operating systems should not be difficult, EPA has not tested and is not prepared to support them. MOVES2004 does not operate successfully with versions of WINDOWS which predate WINDOWS 98 and most WINDOWS 98 systems do not have sufficient resources to run MOVES2004 well.

The MOVES2004 installation package includes, in addition to the MySQL server component needed by MOVES itself, a command line MySQL client program and an installation package for the MySQL Control Center which is a GUI MySQL client program. Either of these MySQL client programs can be used to help construct MOVES2004 input databases and to analyze the contents of MOVES2004 output databases. Other data base management software, such as Microsoft ACCESS can also be used via an ODBC driver. Appendix B of the MOVES2004 User Guide explains how this is accomplished.

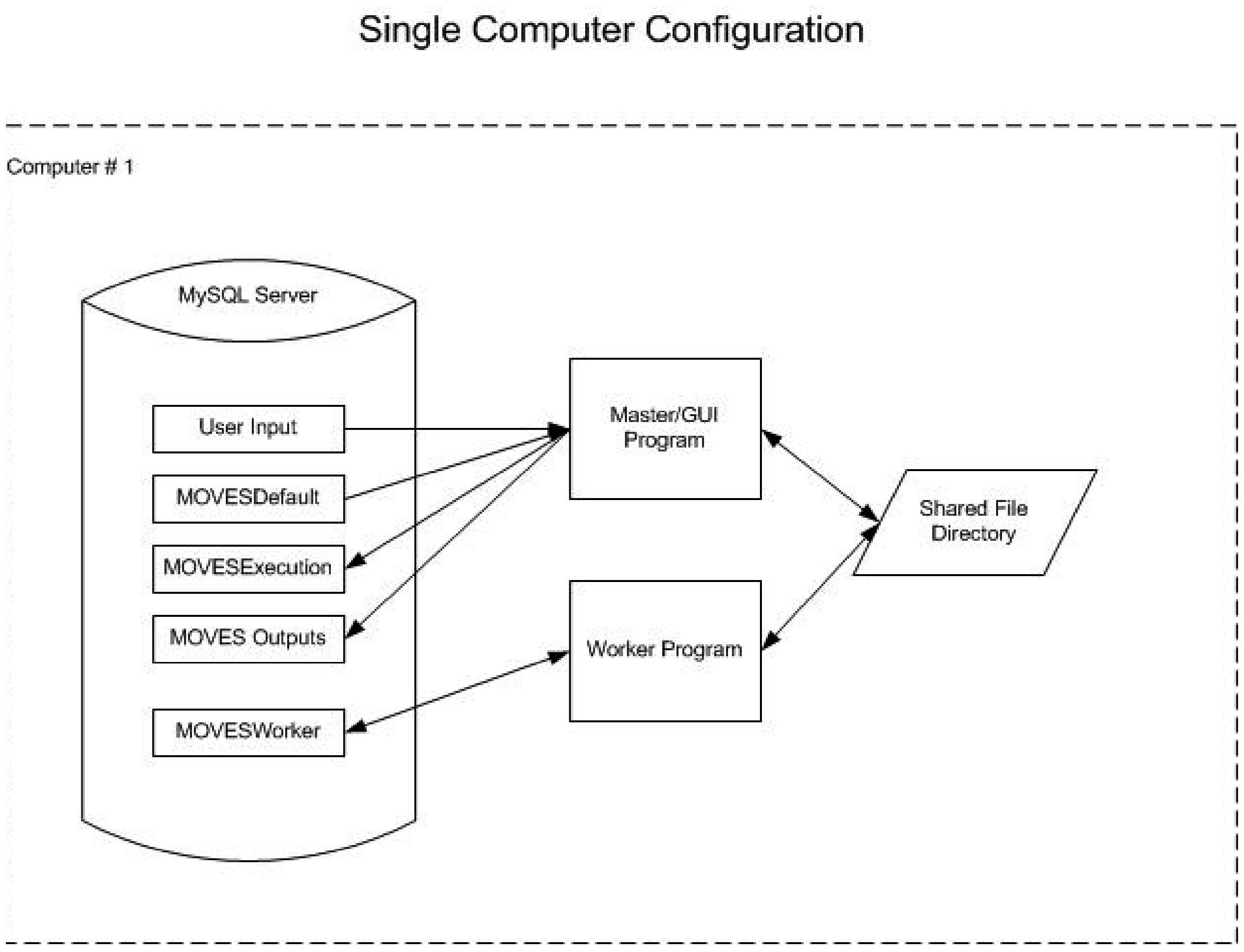
Additional information about MySQL is available at the MySQL web site (http://www.mysql.com/) operated by MySQL AB.

3.3. Details on Shared File Directory Platform Requirements

The SharedWork file directory may be located on a computer that runs the MOVES GUI/Master Program or the MOVES Worker Program or on a separate “file server” computer. All that is necessary is that the MOVES GUI/Master program and at least one MOVES Worker program have access to this shared file directory and permission to create, modify and delete files. In the simplest case, diagrammed early in the next section, all MOVES Application Software components may reside on a single computer. In this case the SharedWork directory is simply on this computer’s local hard drive. All files created by MOVES in this directory are temporary, but because the files can be numerous and large, at least 1GB of disk space should be available.

4. MOVES Computer Platform Configuration

All MOVES2004 application components may be installed on a single computer system as shown in the following Figure 4-1.



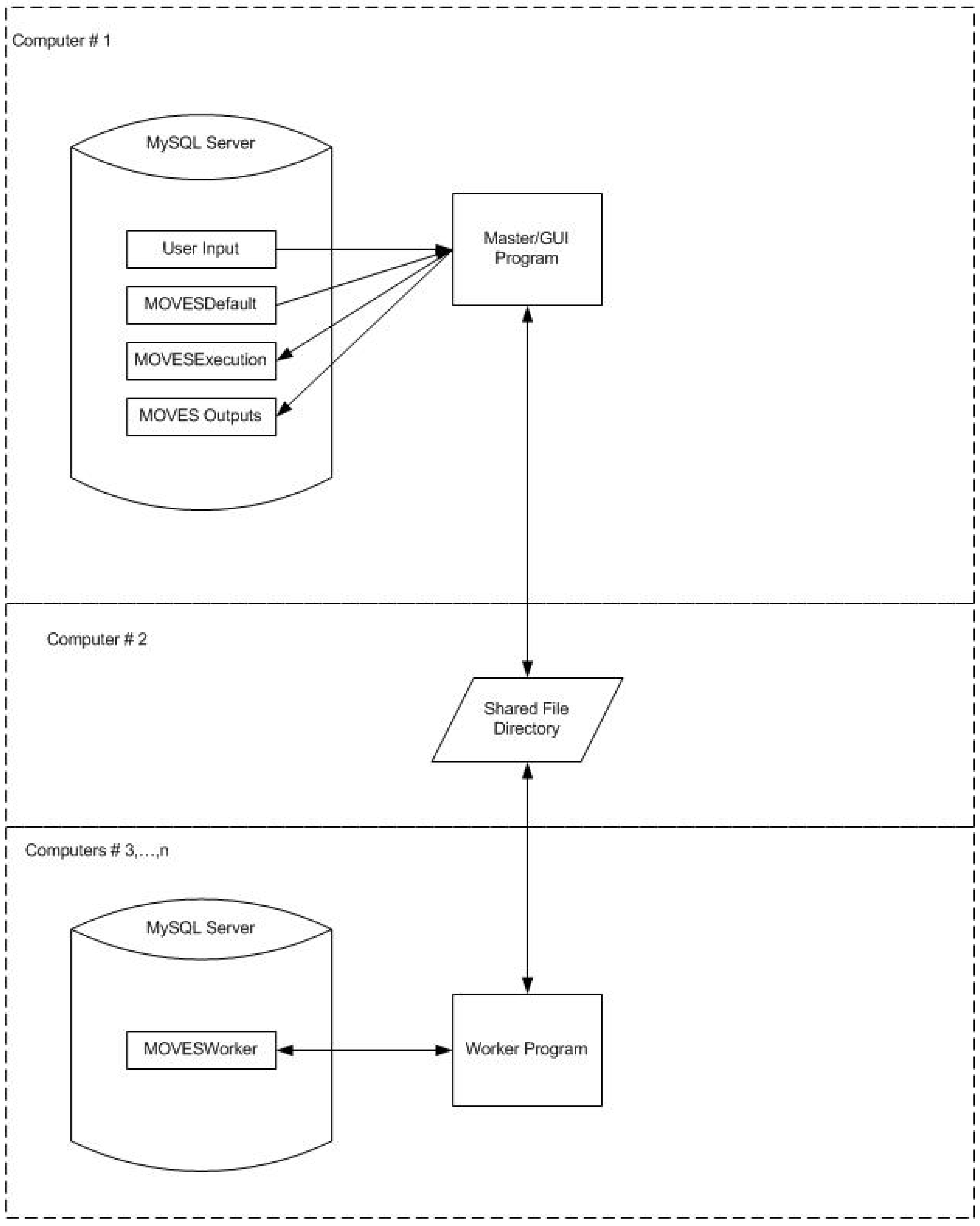
MOVES2004 may also be configured so that several computers work together to execute model simulation runs. This can significantly improve execution time performance of large simulations. (Improvements diminish, however, as more worker computers are added. The number of worker computers needed to approach the minimal execution time for a model run depends on the specific nature of the run.)

All that is necessary for several computers to work together on a MOVES run, is for the computers to have access to the SharedWork directory. For one computer this file directory can be on its local hard drive; other computers must access the SharedWork directory via a computer network. Of course, the platform requirements of each MOVES component must still be satisfied by the computer(s) on which it is installed. A variety of network configurations are possible. The principal consideration is that each Master/GUI Program must have the MOVESDefault database on its computer and that each Worker Program must be able to create a MOVESWorker database on its computer.

Two text files, MOVESConfiguration.txt, and WorkerConfiguration.txt, versions of which are built by the installation program, are used by the MOVES Master/GUI program and the MOVES Worker program to locate their databases and the SharedWork directory.

Figure 4-2 shows a typical Multiple Computer configuration.

Typical Multiple Computer Configuration



5. MOVES2004 Software Licensing

EPA distributes a complete installation package for MOVES2004 as open source software. EPA asserts a copyright to the MOVES2004 application but allows MOVES2004 to be used pursuant to the GNU General Public License (GPL), which is widely used for the distribution of open source software.

Restrictions apply to use of MOVES2004 pursuant to the GPL. For example, the program may not be sold, even in modified form, for commercial profit without obtaining a commercial license to MySQL from MySQL AB and, if redistributed in modified form, must be identified accordingly and source code included. Distribution of modified versions of the program must also require compliance with the GPL unless commercial licenses are obtained. The terms of the GPL are explained in detail at http://www.gnu.org/licenses/.

6. Installation Overview

The MOVES2004 installation package supplied by EPA includes two installation "setup" programs. The first, obtained directly from MySQL AB's web site, is used to install MySQL; the second installs the MOVES2004 application (including its Java source code and compiled class files) along with all other required software platform components.

Before MOVES2004 can be installed it is first necessary to run the MySQL installation set up program on each computer where it is needed in the desired configuration. This installation program is included on the MOVES2004 installation package, and must be run separately, before MOVES2004 itself is installed.

All other MOVES2004 and platform components can be installed by running the MOVES2004 installation. This graphical "wizard-style" program guides the user through the installation process. **Default choices are offered which are suitable for the single computer configuration and for the "typical beginning" case where none of the MOVES platform software is currently installed.** Options are available to install just the MOVES Master/GUI program, or just the MOVES Worker Program, and to specify the location of the SharedWork directory. The approach taken regarding the underlying MOVES software platform components, such as the Java SDK, is to offer the choice of installing them, or to indicate where they have already been installed. One should not rely upon previously installed components, however, unless they are the same version as those in the MOVES Installation package.

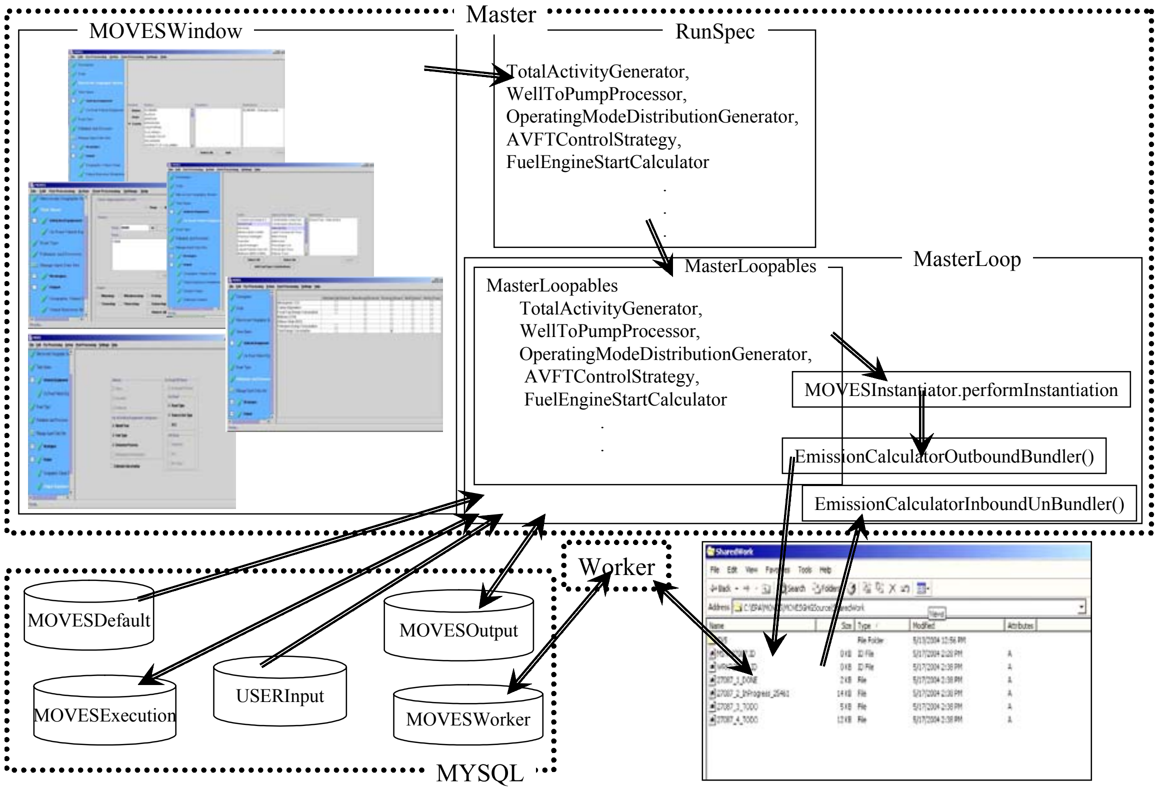
Once the user has become familiar with the components of MOVES, this installation can be modified, in part by editing the configuration files.

The MOVES installation keeps track of components it has installed and includes an "Uninstall" feature to remove those components. It generally cannot be used to remove components which have been modified or which have been installed in some other fashion.

Additional step-by-step detail for installing MOVES2004 is contained in the *MOVES Installation Guide* which should be used when actually performing installation.

7. MOVES2004 Processing Overview

This diagram illustrates the overall flow of processing in MOVES2004 in a way which illustrates the division of work between the MOVES Master and Worker programs.



Typical Processing Steps:

Before beginning a model run any desired User Input datatabases (shown near the bottom left of the diagram) must be prepared. (The diagram does not attempt to show this intial step, which is only required if the user wishes to deviate from the default inputs in MOVESDefault). A button in the MOVES graphical user interface, represented in the diagram by its MOVESWindow, can be used to create an empty User Input database to which users can add the table records they need. The Future Emission Rate Creator external control strategy, discussed later, produces a User Input database for its particular purpose, as does the GREET model interface.

A run specification, or RunSpec, is loaded or produced by using the MOVES graphical user interface (MOVESWindow).

The user initiates execution of the actual model run via the “Action” menu item of MOVESWindow.

The MasterLoop, within the MOVES Master program, then merges any User Input databases identified in the RunSpec with the MOVESDefault database to produce the MOVESExecution database. Most data not needed to fulfill the RunSpec is discarded or “filtered” in this process.

The MasterLoop then uses the MOVESExecution database to produce files containing work “bundles” in the SharedWork directory.

MOVES Worker program(s) perform these bundles of work, using their MOVESWorker databases for temporary storage. They place files containing the completed work back into the SharedWork directory.

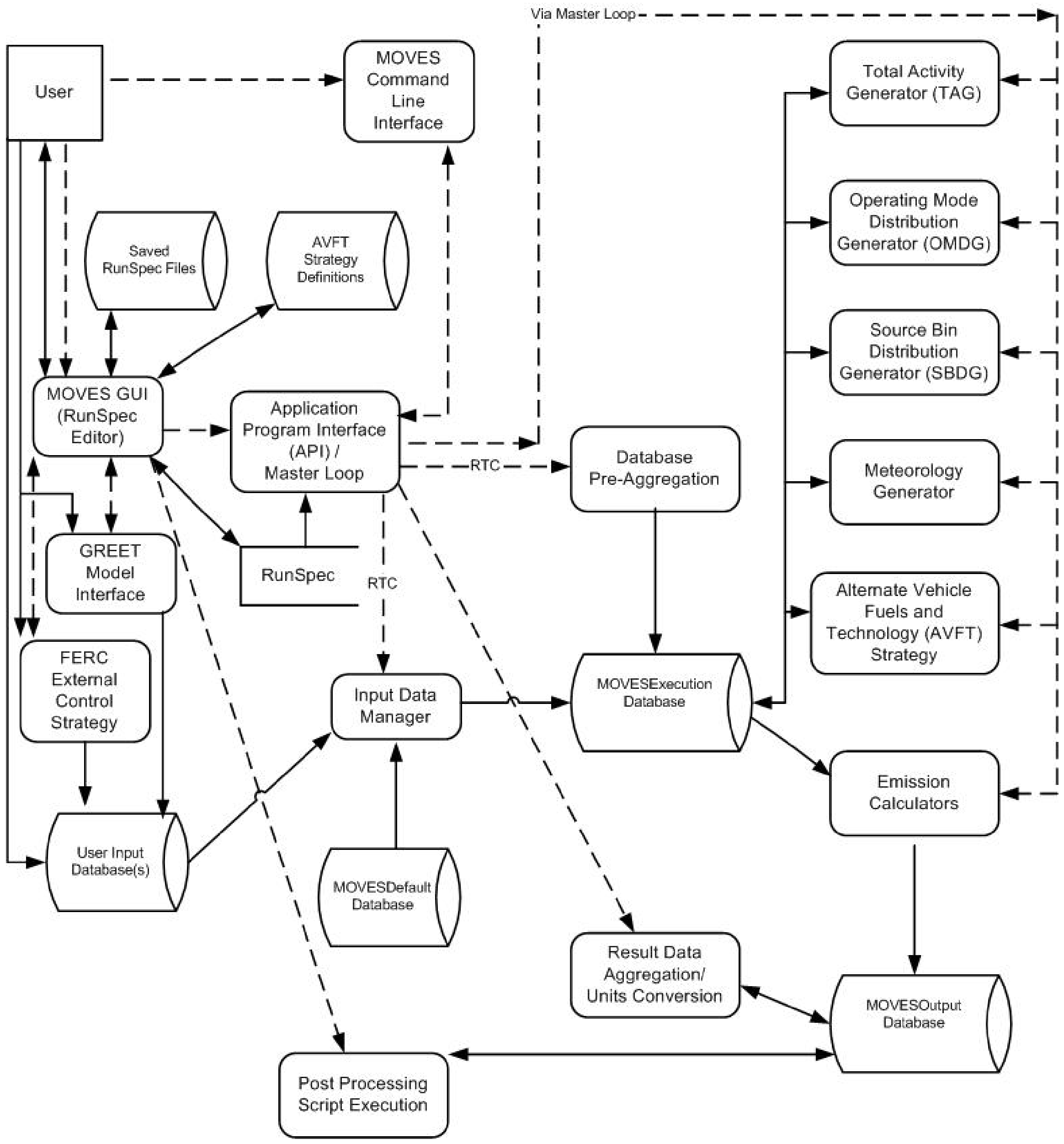
The MasterLoop retrieves these completed work files and processes them into a MOVES output database. The name of the output database is specified in the Run Spec.

The “Post Processing” menu in the MOVESWindow allows for additional, optional processing steps to be performed on the output database.

8. MOVES2004 Data and Control Flow

Figure 8-1 illustrates the logical flow of data and control within the MOVES2004 software. While generally more detailed than the diagram in the previous section, it does not attempt to illustrate the division of work between the MOVES Master and Worker.

Figure 8-1 MOVES Logical Level Data Flow Diagram



Via Master Loop
User
MOVES Command Line Interface
Saved RunSpec Files
AVFT Strategy Definitions
MOVES GUI (RunSpec Editor)
Application Program Interface (API) / Master Loop
RTC
Database Pre-Aggregation
GREET Model Interface
RunSpec
RTC
FERC External Control Strategy
Input Data Manager
MOVESExecution Database
User Input Database(s)
MOVESDefault Database
Total Activity Generator (TAG)
Operating Mode Distribution Generator (OMDG)
Source Bin Distribution Generator (SBDG)
Meteorology Generator
Alternate Vehicle Fuels and Technology (AVFT) Strategy
Emission Calculators
Result Data Aggregation/ Units Conversion
MOVESOutput Database
Post Processing Script Execution

The same graphical conventions are used in this diagram as in Data Flow Design (chapter 5) of the *Design and Implementation Plan for EPA's Motor Vehicle Emission Simulator,* namely*:*

> Cylinders represent databases.
> Rectangles open on the right hand side represent relatively simple data storage elements.
> Round-cornered rectangles represent processes that operate on and produce data.
> Square boxes represent interfaces external to the system.
> Solid line arrows represent data flow.
> Dashed line arrows represent control flow.
> The "RTC" notation on an arrow indicates that the control flow "runs to completion" before any of the data is processed by subsequent steps.

In order to illustrate the specifics of MOVES2004, several databases envisioned by the general MOVES design have been combined into the MOVESDefault and MOVESExecution databases illustrated here. Chapter 11 contains a more detailed discussion of these databases.

The following text attempts to lead the reader through the diagram. The first occurance of each diagram block is bolded.

The **User** normally executes the **MOVES GUI** program, which may access and update files containing **Saved Run Specifications** and **AVFT Strategy Definitions**. A simple **MOVES Command Line Interface** is available as an alternative to execute an existing run specification or **RunSpec**.

The Command Line Interface passes a selected Run Spec directly to the **MOVES Application Program Interface (API)**, whereas the MOVES GUI Program allows the user to select, edit, and save run specifications, and/or operate the **GREET Model Interface** , and/or the **Future Emission Rate Creator (FERC) External Control Strategy**, before passing control to the MOVES API.

Once control is passed to the MOVES API, it manages the remainder of the model run. Control is first passed to the **Input Data Manager,** which merges the

MOVESDefault database with any **User Input Databases** specified by the RunSpec, producing the **MOVESExecution** database.

A **Database Pre-aggregation** function was added to MOVES2004 to reduce execution time performance at the expense of some added approximation in the calculations. This function is executed next if called for by the run specification.

The **Master Loop** then manages execution of the generators, internal control strategies, and calculators. MOVES2004 includes four generators: the **Total Activity Generator (TAG), the Operating Mode Distribution Generator (OMDG**), the **SourceBinDistributionGenerator (SBDG),** and the **Meteorology Generator**. It includes one internal control strategy: the **Alternative Vehicle Fuels and Technologies (AVFT) Strategy.** All generators and control strategies receive input from various portions of the MOVESExecution database and place their output there as well. Tables written by Generators are termed "Core Model Input Tables" or "CMITs" and have an important role in the design of MOVES. Additional detail as to which tables are used by each generator and control strategy is contained in Chapters 10, 11, and 12.

Emission Calculators are the most central or "core" portion of MOVES model. They consume much of its execution time and so the MOVES software design provides for portions of them to run by the MOVES Worker Program. They receive input from the MOVESExecution database (some of which has been produced or altered by the generators and the control strategies). The CMIT tables within the MOVESExecution database provide their principal input data, but other tables may also be used by emission calculators for specialized calculations. The emission calculators output their results to databases on worker machines that are further processed to become the **MOVESOutput Database**. MOVES2004 includes EmissionCalculators for: energy consumption (all processes), distance (even though distance is not an emission), CH4 and N2O – start and running, and CH4 and N2O – well-to-pump.

Result Aggregation and Engineering Units Conversion functions are performed on the MOVESOutput database as final steps of the model run.

Following the model run, the MOVES GUI can be used to invoke additional "post-processing" functions to operate on MOVESOutput databases. MOVES2004 includes one post-processing function**, Post-Processing Script Execution**, which runs a selected MySQL script against the MOVESOutput database specified by the run specification. Several post processing scripts are provided with the model and users may add to these if further customization of MOVES output is desired.

9. MOVES Functional Design Concepts

The functional scope of MOVES2004 can be characterized as follows:

Geography: The entire U.S. (plus Puerto Rico and the U.S Virgin Islands) at the county level, with options to run at a more aggregate state or national level.

Time Spans: Energy/emission output by hour of the day, day of the week, and month for calendar years 1999 through 2050, with options to run at more aggregate day, month or year levels.

Sources: All highway vehicle sources, divided into 13 "use types"

Outputs and Pollutant Emissions: Energy consumption (characterized as total energy, petroleum-based energy and fossil fuel-based energy), N2O, and CH4

Emission Processes: running, start, extended idle (e.g. heavy-duty truck "hoteling"), and well-to-pump (via interface with an updated version of Argonne National Laboratory's GREET model)

Understanding how MOVES2004 operates, and why the input and output databases are set up as they are, requires an understanding of some basic functional design concepts. Fundamental aspects of the MOVES design are geographic locations, time periods, emission sources, emission pollutants, emission processes, vehicle fuels, and emission source activity.

9.1.Geographic Locations

The default geographic modeling domain in MOVES2004 is the entire United States of America. This domain is divided first into "*states*." In this context the District of Columbia, Puerto Rico, and the U.S.Virgin Islands are considered to be "states", so the nation has 53 states in the MOVES database.

"States" are divided into "*counties*." In the MOVESDefault database these correspond to the 3222 political subdivisions of the states in 1999. Counties must belong to a single state. While the county-level subdivisions of some states have changed slightly over time, the MOVES database is set up to store only a single set of counties. The database could be adapted relatively easily to a different set of counties, but it would be a significant structural change to MOVES to allow the set of counties to depend upon the calendar year.

In the general MOVES design framework, "counties" may be further divided into "*zones*," but in MOVES2004 each county is consists of a single zone. Zones are intended to play a distinct role in future versions of MOVES that operate at smaller scales.

"Zones," and thus "counties," in MOVES2004, are further divided into "*links*." In MOVES2004 each county is divided into 13 links, corresponding to the twelve "*road types*" in HPMS, along with an off-network "road type" representing locations not on the county's roadway network. In MOVES2004, running process emissions are considered to occur on the twelve HPMS road types, while its other emission processes (i.e. start, extended idling, and well-to-pump) are associated with the "off network" link locations. (Emission processes are discussed in a subsequent section.)

In MOVES2004, a link is a combination of a road type and a county. Road types, while not in themselves geographic locations, help to define links at the macroscale. In future versions of MOVES, however, when modeling at smaller scales, links may represent segments of actual roadways and road types will serve only to classify links.

Finally, the general MOVES geographic framework envisions that zone locations may be overlaid by a set of "*grids*," or grid cells. There may be multiple grid cells in a zone and multiple zones in a grid cell, but grids play no role in MOVES2004.

9.2. Time Periods

MOVES2004 describes time in terms of calendar years, 12 months of the year, 7 days of the week, and 24 hours of the day. This arrangement appears simple but there are several subtleties to keep in mind.

First, while calendar years in MOVES are intended to represent actual historical years, the finer time period classifications (months, days, and hours) do not represent historical time periods and are best thought of as being "generic" time classifications. Among other things this means that MOVES does not attempt to model the fact that Fridays in December 2004 include a Christmas day holiday, and that in 2005 they do not.

Another reason why MOVES should not normally be considered to model historical time periods smaller than a calendar year is the disconnection between "days of the week" and "months." For example, there is no way to represent data specific to both Monday, May 3, 2004 and Monday May 17, 2004 in a single MOVESDefault or Input Database. The database is only set up to store information about the year 2004 and about Mondays in May. Along the same lines, MOVES does not attempt to model facts such as that there are five Mondays in May, 2004 but only four in June, 2004. MOVES does account for the different number of days in each month (considering 1/7 of them to be Mondays, 1/7 Tuesdays, etc.). MOVES2004 does account for leap years in most respects.

In order to model an actual historical time period, data corresponding to the unique time period would have to be supplied by the user. Depending on the accuracy and detail desired, multiple model runs might be necessary if the time period were longer than seven days.

9.3. Characterizing Emission Sources (Vehicle Classification)

A long-standing challenge in the generation of on-road mobile source emission inventories is the disconnect between how vehicle activity data sources characterize vehicles and how emission or fuel economy regulations characterize vehicles. The crux of this issue is that there is a fundamental difference between factors influencing how vehicles are used, and their fuel consumption and emission performance. An example of this is how vehicles are characterized by the Highway Performance Monitoring System (HPMS) – by a combination of the number of tires and axles – and EPA's weight-based emission classifications such as LDV, LDT1, LDT2 etc.

This disconnect is fundamental to matching activity data and emissions data, and generally requires some "mapping" of activity data to emission data. The MOBILE series of models have traditionally grouped vehicles according to the EPA emission classifications, and provided external guidance on mapping these categories to the sources of activity data, such as HPMS. MOVES is designed to take these mappings into account internally, such that the casual user of MOVES will not have to deal with external mapping. Doing this, however, requires some complexity in the design. Vehicles are characterized both according to activity patterns and energy/emission performance, and are mapped internal to the model. Thus the model uses data for both the activity and energy/emission methods of characterization. On the activity side, vehicles are grouped into "*Source Use Types*," or "*Use Types"*, which are expected to have unique activity patterns. Because the HPMS is a fundamental source of activity information, the MOVES use types are defined as subsets of the HPMS vehicle classifications. These use types are shown in Table 9-1.

Table 9-1. MOVES Source Use Type Definitions

HPMS Class	MOVES use type	Description
Passenger Cars	21. Passenger Car	
Other 2-axle / 4-tire Vehicles	31. Passenger Truck	Minivans, pickups, SUVs and other 2-axle / 4-tire trucks used primarily for personal transportation
	32. Light Commercial Truck	Minivans, pickups, SUVs and other trucks 2-axle / 4-tire trucks used primarily for commercial applications. Expected to be unique from passenger trucks in terms of annual mileage, operation by time of day
Single Unit Trucks	51. Refuse Truck	Garbage and recycling trucks Expected to be unique from other single unit trucks in terms of drive schedule, roadway type distributions, operation by time of day
	52. Single-Unit Short-Haul Truck	Single-unit trucks with majority of operation within 200 miles of home base
	53. Single-Unit Long-Haul Truck	Single-unit trucks with majority of operation outside of 200 miles of home base
	54. Motor Home	
Buses	41. Intercity Bus	Buses used primarily by commercial carriers for city-to-city transport.
	42. Transit Bus	Buses used within an urban area
	43. School Bus	
Combination Trucks	61. Combination Short-Haul Truck	Combination trucks with majority of operation within 200 miles of home base
	62. Combination Long-Haul Truck	Combination trucks with majority of operation outside of 200 miles of home base
Motorcycles	11. Motorcycle	

Activity patterns which may differ between the use types are: annual mileage, distribution of travel by time of day or day of week, driving schedule (i.e. real time speed/accel profile), average speeds, or distribution of travel by roadway type. For example, refuse trucks are separated out because their activity patterns are expected to vary significantly from other single-unit trucks, and accurately accounting for these vehicles requires accounting for their unique activity.

Source use types are the principal method of vehicle characterization seen by the MOVES user. The user selects which use type and fuel combinations to model in the user interface, and results are best reported by use type.[1] However, emission rates contained in the model are not broken down by use type, for two (related) reasons: first, emission and fuel consumption data are not gathered according to use types or other activity-based classifications (e.g. HPMS). Second, the factors that influence fuel consumption and emission production are different from how vehicles are used. For example, with regard to fuel consumption, loaded vehicle weight is a predominant influence; a 2000 lb. compact car and 5000 lb. SUV will have very different fuel consumption levels, although these vehicles may have similar use patterns. It is necessary to account for these differences in fuel consumption and emission generation, separate from activity pattern. To do this, the MOVES design has implemented the concept of "*Source Bins*." Unique source bins are differentiated by characteristics that significantly influence fuel (or energy) consumption and emissions – and because these vary by pollutant, they are allowed to vary by pollutant in MOVES. Table 9-2 shows the source bins fields used in MOVES2004, which vary by pollutant. Energy source bins are defined by fuel type, engine type, model year group, loaded weight and engine size. For CH_4 and N_2O, source bins are defined by fuel type, engine type, model year group (notice the definition of model year group can vary by pollutant, which is necessary to account for trends which vary by pollutant), and regulatory class.

[1] Because Source Classification Codes (SCCs) have been and are used extensively in emission inventories MOVES2004 also offers the option of reporting results by SCC. There are currently 144 SCCs for mobile sources formed by the intersection of the 12 HPMS road types with the 12 vehicle classifications used in PART5 and NMIM. This scheme is not native to MOVEs, however, and the MOVEs modeling team would like to discourage continued use of this classification scheme.

Table 9-2. MOVES Source Bin Definitions

Fuel Type (Energy, CH_4, N_2O)	**Engine Technology** (Energy, CH_4, N_2O)	**Model Year Group**		Loaded Weight (Energy)	**Engine Size** (Energy)	Regulatory Class (CH_4, N_2O)
		Energy	(CH_4, N_2O)			
Gas Diesel CNG LPG Ethanol (E85) Methanol (E85) Gas H_2 Liquid H_2 Electric	Conventional IC (CIC) Advanced IC (AIC) Hybrid - CIC Moderate Hybrid - CIC Full Hybrid - AIC Moderate Hybrid - AIC Full Fuel Cell Hybrid - Fuel Cell	1980 and earlier 1981-85 1986-90 1991-2000 2001-2010 2011-2020 2021 and later	1972 and earlier 1973 1974 1975 . . . 1999 2000 2001-2010 2011-2020 2021 and later	Null < 500 (for motorcycles) 500-700 (for motorcycles) > 700 (for motorcycles) <= 2000 lbs 2001-2500 2501-3000 3001-3500 3501-4000 4001-4500 4501-5000 5001-6000 6001-7000 7001-8000 8001-9000 9001-10,000 10,001-14,000 14,001-16,000 16,001-19,500 19,501-26,000 26,001-33,000 33,001-40,000 40,001-50,000 50,001-60,000 60,001-80,000 80,001-100,000 100,001-130,000 >=130,001	Null < 2.0 liters 2.1-2.5 liters 2.6-3.0 liters 3.1-3.5 liters 3.6-4.0 liters 4.1-5.0 liters > 5.0 liters	Null Motorcycle LDV LDT HDT

Source bins are defined independently from use types, but are mapped to use types internal to MOVES by the SourceBinDistributionGenerator.

9.4. Emission Pollutants

MOVES2004 estimates two fundamentally different kinds of results: energy consumption and mass emissions. For convenience, all these quantities are considered to be "emissions." "Energy emissions" estimated by MOVES2004 are total energy consumption, fossil fuel energy consumption, and petroleum fuel energy consumption. The more familiar mass emissions estimated by MOVES2004 are methane (CH4) and nitrous oxide (N2O.

9.5. Emission Processes

On-road vehicles consume energy and produce mass emissions through several mechanisms or pathways, which are known within MOVES as "*emission processes*," or just "*processes*," and are accounted for and reported (if desired by the user) separately. The MOVES2004 mechanisms for "pump-to-wheel" energy consumption and gaseous emissions are limited to operation of the engine and emissions from the tailpipe. However, certain aspects of engine operation are so different in terms of their activity and affect on exhaust emissions, they merit treatment as separate exhaust emission processes. In addition to these "pump-to-wheel" exhaust emission processes MOVES2004 also includes a "well-to-pump" emission process. The processes for MOVES2004 are as follows:

Running Exhaust, meaning the energy consumed or the tailpipe emissions produced during vehicle operation over freeways and surface streets while the engine is fully warmed up.

Start Exhaust, meaning the additional energy consumed or tailpipe emissions produced during the period immediately following vehicle start-up. An important note is that this quantifies the energy consumed or emissions produced *in addition* to the "running" energy/emissions produced immediately following start-up. Start emissions represent the incremental emissions produced following vehicle start-up, after accounting for the baseline running emissions.

Extended Idle, meaning energy consumed during long periods of engine idling off of the roadway network. This process applies mainly to combination long-haul trucks in MOVES, and is meant to account for the emerging issue of overnight "hoteling" at truck stops, although it could eventually be applied to idling of passenger vehicles in drive-thru lanes, etc. This process is implemented only for energy consumption in MOVES2004.

Well-To-Pump, meaning the energy and emissions produced from processing and distributing vehicle fuel in the process of getting from raw feedstock to the fuel pump. These energy use and emission rates are produced by Argonne National Laboratory's GREET model.

It should be noted that well-to-pump emissions have a different relationship to locations than those of the other processes. MOVES2004 associates well-to-pump results with the locations (e.g. Counties and RoadTypes) whose activity *caused* the emissions; the emissions *are not produced* at these locations as they are for the other processes.

An additional process, manufacture/disposal, would account for energy and emissions from vehicle production and disposal. This is not ready for inclusion in MOVES2004, but is planned for inclusion in a later version.

9.6. Vehicle Fuel Classifications

The top level vehicle fuel classifications are listed in Table 9-2, above, in the context of their role in vehicle source bin classification. Implicit in this source bin classification scheme is that a particular vehicle is designed to operate on one of these top level fuel types (e.g. gasoline, diesel fuel, electricity, etc.)

To facilitate modeling the effects of alternative fuels, MOVES2004 further divides these top level fuel types into fuel subtypes. In MOVESDefault, for example, the gasoline fuel type has three subtypes: conventional, reformulated, and gasohol (E10). Diesel fuel has three subtypes: conventional, biodiesel, and Fischer-Tropsch diesel. Fuel subtypes represent alternative ways of meeting the demand for a general type of fuel. MOVES2004 assumes that vehicles designed to operate on a top level fuel type may be operated on any of its subtypes depending upon the fuel supply at a particular time and geographic location.

9.7. Emission Source Activity

The cornerstone of estimating mobile source energy usage and emission inventories is vehicle activity. Vehicle activity centers on two fundamental questions: what is the total amount of vehicle activity, and how is this activity subdivided into modes that are unique to energy consumption and emissions. The first question is quantified in MOVES by the metric Total Activity. Total Activity, as the name implies, is the total amount of vehicle activity for source use types in the given location and time which the user has selected in the run specification. The basis of total activity depends on the emission process, as shown in Table 9-3. In MOVES2004 the Total Activity Generator estimates Total Activity for the Start Exhaust, Running Exhaust and Extended Idle emissions processes.

Table 9-3. Total Activity Basis by Process

Emission Process	Total Activity Basis	Description
Running	Source Hours Operating (SHO)	Total hours, of all sources within a source type, spent operating on the roadway network for the given time and location of the run spec. The same as number of sources * per-source hours operating
Start	Number of Starts	Total starts, of all sources within a source type, for the given time and location of the run spec. The same as number of sources * per-source starts
Extended Idle	Extended Idle Hours	Total hours, of all sources within a source type, spent in extended idle operation for the given time and location of the run spec.
Well-To-Pump	Pump-To-Wheel Energy Consumed	Total energy consumed, of all sources within a source type, for the given time and location of the run spec. The sum of running, start and extended idle.

The second piece of activity characterization is to define how total activity is subdivided into operating modes which produce unique energy consumption and emission rates. In the MOVES design these operating modes are allowed to vary by emission process and even pollutant, although for MOVES2004, discrete operating modes are only employed for running energy consumption based on combinations of vehicle specific power (VSP) and instantaneous vehicle speed. These operating modes are shown in Table 9-4. The operating mode concept is central to MOVES multi-scale analysis capability, and will be expanded with the criteria pollutant version of the model.

Table 9-4. Operating Mode Bin definitions for MOVES2004 (running energy consumption)

Braking (Bin 0)			
Idle (Bin 1)			
VSP \ Speed	**0-25mph**	**25-50**	**>50**
< 0 kW/tonne	Bin 11	Bin 21	-
0 to 3	Bin 12	Bin 22	-
3 to 6	Bin 13	Bin 23	-
6 to 9	Bin 14	Bin 24	-
9 to 12	Bin 15	Bin 25	-
12 and greater	Bin 16	Bin 26	Bin 36
6 to 12	-	-	Bin 35
< 6	-	-	Bin 33

10. MOVES Functional Specifications

This chapter explains the functions performed by each portion of the MOVES software, or indicates where such information can be found.

10.1. MOVES Graphical User Interface (GUI) / Run Specification Editor

The MOVES Graphical User Interface (GUI) is used to produce and modify MOVES run specifications. Its functionality is described in detail in the MOVES2004 User Guide.

10.2. MOVES Application Program Interface and Master Looping Mechanism

This component, shown in figure 8-1, manages the overall execution of a MOVES model run. It includes an application program interface (API) callable by either the command line interface, or the MOVES GUI. It invokes the input data manager and any database pre-aggregation.

This component includes a master looping mechanism. InternalControlStrategy, Generator, and EmissionCalculator objects "sign up" with this MasterLoop to execute over portions of the modeling domain (the modeling domain being defined by a RunSpec and being divisable by geography, time, and emission process).

This component creates and manages a pool of control "threads" to bundle Calculator input data (and SQL scripts to be run on the data) for portions of the modeling domain and places them in the SharedWork directory. It also includes a thread of control to unbundle the results placed in the SharedWork directory by MOVES Worker program(s).

This software component is rather complex and consists of a number of Java classes. Programmer level documentation is required to understand this component in greater detail.

10.3. Input Data Manager

The InputDataManager generates the MOVESExecution database from the MOVESDefault database, removing (or "filtering") records in this process based on the needs of the run specification (RunSpec). The InputDataManager runs to completion before any InternalControlStrategy objects, Generators or Calculators begin the MOVES model calculations. The MOVES GUI and RunSpecs specify a list of input databases to be used in addition to the default database to construct the MOVESExecution database.

The InputDataManager filters input records based on the following RunSpec criteria: year, month, link, zone, county, state, pollutant, emission process, day, hour, day and hour combination, roadtype, pollutant-process combination, sourceusetype, fueltype, fuelsubtype, and monthgroup. This filtering is specified in a table-specific manner and need not be applied to every table having these key fields. Filtering is not performed on particular table columns where doing so would interfere with correct result calculation. (The exact table columns to filter are specified in the Java code for the InputDataManager class in the “tablesAndFilterColumns” array.) The reasons for not filtering a particular table by all possible criteria are documented with program source code comments.

Input databases have the same table contents and structure as the MOVESDefault database, but need not contain all tables. If a table is present, however, it must contain all the table columns. Records from user input databases add to or replace records in the MOVESDefault database. If the same record (i.e. a record having the same values of all primary key fields but generally different non-key field values) is present in more than one input database, the record from the user input database listed last in the GUI and in the RunSpec is the one which ends up in MOVESExecution.

A field is added to core model input tables in the MOVESExecution Database as they are generated by the Input Data Manager to indicate that the record came from an input database so that Generators may avoid deleting such records.

MOVES verifies that all input tables contain the required columns. The MOVES GUI also checks that the same database is not specified for both input (either as the default input database or an additional input database) and for output, and ensures that MOVESExecution is not used as either an input or an output database. Otherwise, however, it remains the responsibility of the user to ensure that the ordered application of any additional input databases called for in the run specification to the MOVESDefault database results in a MOVESExecution database that is accurate, complete and consistent.

10.4. Database Pre-Aggregation

To improve execution run time performance, when geographic selections are made at the state or national level, MOVES "preaggregates" the MOVESExecution database so that each state selected, or the entire nation, appears in the database as if it were a single county. These geographic performance shortcuts are specified by the "STATE" and "NATION" GeographicSelectionType values produced by the "Macroscale Geographic Selection" GUI screen and stored in MOVES run specifications. No database preaggregation is performed when geographic selections are made at the County level. County selections may still be used to produce results for broad geographic areas if the user can endure their execution time performance.

Options are also available to improve execution run time performance by "preaggregating" time periods in the MOVESExecution database. These options are specified by the "Time Aggregation Level" item in the MOVES GUI and MOVES RunSpec. This can assume the following values:

HOUR - no time period aggregations are performed

DAY - combine all hours of the day

MONTH - combine all days of the week

YEAR - combine months of the year

All of these computational shortcuts (except COUNTY and HOUR) involve compromises to the accuracy of the results.

The MOVES GUI adjusts the levels of geographic and time period detail specified for the output if necessary so that levels of output detail which can no longer be produced due to data preaggregation are not requested by the RunSpec.

10.4.1. Sequence of the Database Pre-Aggregation Operations:

After creation of the MOVESExecution database by the input data manager the geographic and time period preaggregation operations are performed as follows:

a. If the GeographicSeletionType = NATION, the model creates an average county which represents the entire nation. To do this the MOVESExecution database is aggregated to a level where the nation consists of a single representative state and this “state” consists of a single “county”, and therefore a single “zone”. There is a single “link” for each road type in the RunSpec.

b. If the GeographicSeletionType = STATE, then the MOVESExecutionDatabase is aggregated to a level where each state selection in the RunSpec consists of a single “county” and therefore a single “zone”. There is a single “link” in each such state for each road type in the RunSpec.

c. if the Time Aggregation Level value is DAY, MONTH, or YEAR, all data pertaining to the 24 separate hours of the day in the MOVESExecution database is aggregated into a single “hour” representing the entire day.

d. if the Time Aggregation Level value is MONTH, or YEAR, all data pertaining to the 7 separate days of the week in the MOVESExecution database is further aggregated into a single “day” representing the entire week.

e. if the Time Aggregation Level value is YEAR, all data pertaining to the 12 separate months of the year in the MOVESExecution database is still further aggregated into a single “month” representing the entire year.

f. Following any of the pre-aggregation operations performed in steps a. thru e. the set of ExecutionLocations used by the MasterLoop is recalculated based on the aggregated database.

g. If the DAY, MONTH, or YEAR aggregations have been performed all information derived from the run specification used throughout the remainder of the run is made consistent with the aggregated time periods.

These operations run to completion before any MOVES MasterLoopable objects (ControlStategy objects, Generators, and EmissionCalculators), are invoked.

10.4.2. How the Pre-aggregated Results are Reported

If either of the geographic computational shortcuts is taken, the output database produced does not contain any “real” county (or perhaps even “real” state) level detail, even though such detail is generally present in the MOVESDefault and user input databases. Instead, additional "pseudo" values of stateID, countyID, etc. appear in the output records when a geographical computational shortcut is taken.

If any one of the time period calculation shortcuts is taken, there may be only single representative "hour", "day" or "month" time periods for the MasterLoop to loop over, (though no MasterLoopable objects currently sign up at these low levels), and the output database produced may not contain any "real" hour, day, or month level detail, even though such detail will generally be present in the MOVESDefault and user input databases. Instead "pseudo" time period-identifying values will now be present in the MOVEExecution and MOVESOutput databases.

10.4.3. Algorithms Used to Perform the NATION and STATE Pre-Aggregations

Table 10-1 describes the database aggregation algorithms used on a table-by-table basis for the NATION and STATE cases. Tables not listed contain no geographic identifiers and are therefore not affected by these aggregations. While some of these table aggregations are simple summations, others are "activity-weighted". For these activity-weighted summations to be performed entirely correctly, something approaching the full execution of the control strategies and generators would have to be performed which would defeat the purpose of the pre-aggregation. So, instead, these "activity-weighted" aggregations involve compromise and simplification. Specifically, the activity weighting is based entirely upon "startAllocFactor" values in the Zone table. The variable startAllocFactor is the factor used within MOVES to allocate the total number of starts from the national to the county / zone level. In MOVES2004, this allocation is based on vehicle miles traveled (VMT), hence the use of startAllocFactor for the pre-aggregation weightings is in essence a VMT weighting. More details on startAllocFactor and its derivation can be found in the report "MOVES2004 Highway Vehicle Population and Activity Data".

Table 10-1. Database Aggregation Algorithms

MOVES Database Table	GeographicSelectionType =NATION	GeoGraphicSelectionType =STATE

State	Single Record, stateID=0,
	stateName=Nation,
	stateAbbr=US
		No action required. Already filtered by stateID

County	Single Record, countyID = 0.
	stateID=0, countyName=Nation,
	altitude=L (not used yet)
		Single record per stateID,
		countyID=stateID*1000,
		countyName = stateName,
		altitude = L (not used yet)

Zone	Single Record,
	zoneID=0, countyID=0.,
	startAllocFactor = 1.0
	idleAllocFactor = 1.0
		Single record per stateID,
		zoneID=stateID*10000,
		countyID= stateID*1000
		startAllocFactor = sum of old factors for state.
		Same for idleAllocFactor.

Link	Single record for each roadTypeID in RunSpec.
	linkID=roadTypeID,
	countyID = 0.
	zoneID = 0.
	linkLength = NULL
	linkVolume = NULL
	grade = weighted national average
		Single record for each stateID - roadTypeID in RunSpec.
		linkID= stateID * 100000 + roadTypeID,
		countyID = stateID*1000,
		zoneID = stateID*10000
		linkLength = NULL
		linkVolume = NULL
		grade = weighted state average

CountyYear	Single record for each yearID, countyID = 0	Single record per stateID - yearID combination. countyID = stateID*1000.
ZoneMonthHour	Single record for each monthID-hourID combination in old table. (These have been filtered.) Calculate activity-weighted national average temperature and humidity. heatIndex is recalculated by Met generator.	Single record for each combination of new zoneID, month and hour. (These have been filtered.) Calculate activity-weighted state averages from old county temperature and humidity data heatIndex is recalculated by Met generator.
OpMode Distribution	(contents would be from user input. Warns if data present for some links but not all, for any sourceTypeID-hourDayID-polProcessID-RoadtypeID.) Aggregate to national level roadType linkIDs, weighting by activity.	(contents would be from user input. Warns if data present for some links but not all, for any sourceTypeID-hourDayID-polProcessID- RoadtypeID.)) Aggregate to state level roadType linkIDs, weighting by activity.
ZoneRoadtype	Single record for each roadTypeID. SHOAllocFactor = sum of old SHOAllocFactors	Single record for each new zoneID, roadtypeID combination. SHOAllocFactor = sum of old SHOAllocFactors
IMOBD Adjustment (Default values are considered.)	Activity-weighted aggregation of all old counties to single new county.	Activity-weighted aggregation of all old counties within state.

FuelSupply (Default values are considered.)	Activity-weighted aggregation of all old counties to single new county.	Activity-weighted aggregation of all old counties to single new county for each state.
SHO	(contents would be from user input. Warns if data present for some links but not all for any sourcetype-hourday-month-year-age-roadtype) Combine all links having same roadtype. SHO is a simple summation.	(contents would be from user input. Warns if data present for some links but not all for any sourcetype-hourday-month-year-age-roadtype) Combine all links within state having same roadtype. SHO is a simple summation.
Starts	(contents would be from user input. Warn if data present for some zones but not all for any sourcetype-hourday- month-year-age.) Combine all zones. starts field is a simple summation.	(contents would be from user input. Warn if data present for some zones but not all for any sourcetype-hourday- month-year-age.) Combine all zones within state. starts field is a simple summation.
Extended IdleHours	(contents would be from user input.) Combine all zones. extendedIdleHours field is a simple summation.	(contents would be from user input.) Combine all zones within state. extendedIdleHours field is a simple summation.

10.4.4. Algorithms Used to Perform the Time Period Pre-Aggregations

Table 10-2 describes the database aggregation algorithms used on a table-by-table basis for the DAY, MONTH, and YEAR time period pre-aggregations. These must operate correctly whether or not one of the STATE or NATION aggregations has been performed. Tables not listed contain no time period identifiers and are therefore not affected by these aggregations.

While some of these table aggregations are simple summations, others are “activity-weighted”. All activity-based weighting is in essence based on VMT, using allocations of VMT at the level necessary for the desired aggregation. For these activity-weighted summations to be performed entirely correctly, something approaching the full execution of the control strategies and generators would have to be performed which would defeat the purpose of the pre-aggregation. So instead these “activity-weighted” aggregations involve some compromise and simplification. Specifically, the activity weighting used for the DAY aggregation is based upon the values in the HourVMTFraction table; the weighting used for the MONTH aggregation is based upon the values in the DayVMTFraction table; and the activity weighting used for the YEAR aggregation is based upon the values in the MonthVMTFraction table. Because these activity fractions themselves depend upon other dimensions of the model, which do not always appear in the tables being aggregated, several variations of each aggregation are utilized, some of which are approximations:

For aggregating hours into Days, three activity-weighting variations are used. (The third is an approximation.)

HourWeighting1: is based directly on the HourVMTFraction table itself, which is used to aggregate tables sharing its sourceTypeID, roadTypeID, and dayID primary keys.

HourWeighting2: uses RoadTypeDistribution to aggregate HourVMTFraction over Roadtype. This is used to aggregate tables having sourceTypeID and dayID, but not roadTypeID.

HourWeighting3: is a simple weighting by hourID only used to aggregate tables sharing no keys with HourVMTFraction except hourID. It is produced from HourWeighting2 by using the data for the passenger car sourceUseType, and giving equal weight to all seven days of the week.

For aggregating days into Months, two activity-weighting variations are needed:

DayWeighting1: is based on the DAYVMTFraction table, with MonthID removed by using weights from MonthVMTFraction.

DayWeighting2: is based on DayWeighting1, with RoadTypeID removed by using information from RoadTypeDistribution.

For aggregating months into Years, only one activity-weighting is needed, but it is an approximation:

MonthWeighting1: based on MonthVMTFraction information for passenger cars only.

Table 10-2. Description of Time Period Aggregations to Be Performed

MOVES Database Table	Aggregation of Hours to DAY	Aggregation of Days to MONTH (assumes DAY aggregation.)	Aggregation of Months to YEAR (assumes MONTH aggregation.)
HourOfAnyDay	Single Record, hourID=0, hourName = "Entire day".		
DayOfAnyWeek		Single record. dayID=0. dayName = "Whole Week"	
HourDay	Record for each dayID, hourID=0	Single record. dayID=0. hourID=0.	
MonthOfAny Year			Single record. MonthID = 0 noOfDays = 365 monthGroupID=0
MonthGroup OfAnyYear			Single record. MonthGroupID=0 monthGroupName = "Entire Year".
HourVMT Fraction	Record for each SourceType- RoadType-Day combination. HourVMTFraction = 1.0	Record for each SourceType- RoadType combination. HourVMTFraction = 1.0	
DayVMT Fraction		Record for each SourceType-Month-RoadType combination. DayVMTFraction = 1.0	Record for each SourceType-RoadType combination. DayVMTFraction = 1.0
MonthVMT Fraction			Two records for each sourceTypeID, month VMTFraction = 1.0
AvgSpeed Distribution	Activity-weighted average avgSpeedFraction using HourWeighting1	Activity-weighted average avgSpeedFraction using DayWeighting1	
OpMode Distribution (contents would be from user input.)	Activity-weighted average opModeFraction using HourWeighting1	Activity-weighted average opModeFraction using DayWeighting1	

SourceType Hour	Activity-weighted average startsPerSHO and idleSHOFactor using HourWeighting2	Activity-weighted average startsPerSHO and idleSHOFactor using DayWeighting2	
SHO (contents would be from user input.)	Simple summation of SHO and distance	Simple summation of SHO and distance	Simple summation of SHO and distance
Starts (contents would be from user input.)	Simple summation of starts	Simple summation of starts	Simple summation of starts
Extended IdleHours (contents would be from user input.)	Simple summation of extendedIdleHours	Simple summation of extendedIdleHours	Simple summation of extendedIdleHours
ZoneMonthHour	Activity-weighted average temperature and relHumidity using HourWeighting3.		Activity-weighted average temperature and relHumidity using MonthWeighting1.
MonthGroup Hour	Activity-weighted ACActivity terms using HourWeighting3.		Activity-weighted ACActivity terms using MonthWeighting1.
FuelSupply (Note default values.)			Activity-weighted marketshare using MonthWeighting1.

10.4.5. Calculation Inaccuracies Introduced by the Database Pre-Aggregations:

The simplified activity-weighted aggregations introduce the following approximations relative to having MOVES perform its calculations individually for each real county-location and for each hour of the day:

- Start-based activity, while appropriate for the start process, represents an approximation for other processes whose activity basis (SHO, etc.) may not exactly correspond to start activity.
- Direct user input to the CMIT tables may override the Zone.startAllocFactor values. Any effect on activity of direct user input to the CMIT tables is not taken into account.
- Control strategies may eventually be added to MOVES which adjust activity levels. Any such effects are not included.
- The potentially significant non-linear relationships of the emissions calculations to temperature and humidity are ignored. This may be especially serious at the national level.

- Activity weighted hourly averages are used when combining hourly temperature, humidity, and AC activity information for the hours of the day, but differences in hourly activity levels between the passenger car source use type and other source use types are ignored in this calculation, as are differences in hourly activity levels by day of the week.

- Differences in monthly activity patterns between passenger cars and other source use types are ignored when calculating weighted average annual temperature, humidity, ACActivity, and fuelSubtype marketshares.

- When calculating annual data from monthly data, all years are considered to be non-leap years.

An initial analysis of the sensitivity of MOVES2004 results to levels of pre-aggregation is presented in the report "MOVES2004 Validation Results". While MOVES2004 does produce different results depending on the level of aggregation selected by the user, the magnitude of difference does not appear to be very large. For example, the difference in total energy results between a run where state / month pre-aggregation was selected was about 2 percent higher than the same run where nation / year pre-aggregation was selected.

10.5. Total Activity Generator (TAG)

This generator calculates the total operating activity pursuant to the run specification for each source use type in MOVES2004 using the MOVESExecution database. This operating activity includes Start activity (number of starts) and Extended Idle Hours in addition to source hours operating (SHO), by time span, geographic location, source type and age. The product of the TAG is three core model input tables produced in MOVESExecution containing these results: SHO, Starts, and ExtendedIdleHours. It also includes distance traveled information which is stored in the SHO table.

The algorithm used to calculate total operating activity for a given time and location by source type and age is divided into ten steps (numbered 0 thru 9) referred to as TAGs (total activity generator steps) in this document. Each TAG step references the MOVESExecution database and implements a simple mathematical formula. The primary function of the TotalActivityGenerator is to convert the commonly-available activity data such as vehicle miles traveled (VMT), age distribution, vehicle populations, sales and VMT growth rates, etc. to the MOVES activity parameters of source hours operating, starts, and extended idle hours. Externally-provided VMT is thus the primary driver of total activity. Steps 1-3 exist to calculate allocations of total VMT by vehicle age and source type. Step 4 grows VMT. Steps 5 and 6 allocate VMT by time, space, vehicle age and source type. Step 7 converts the allocated VMT to the MOVES activity parameters of source hours operating (SHO), number of starts, and extended idle hours. Step 8 allocates activity to zones (e.g. counties for the default case), and Step 9 re-calculates distance for output reporting purposes. A brief description of each of the TAGs is shown in table 10-3. The HPMS acronym used in the table refers to the Highway Performance Management System.

Table 10-3. Overview of TAG Calculations

Step		**Description**
TAG-0	Determine the base year	Determine the appropriate base year to use in growth calculations
TAG-1	Calculate Base Year Vehicle Population	Establish base-year vehicle population in modeling domain by use type, age
TAG-2	Grow Vehicle Population to Analysis Year	If analysis year is in future, establish analysis year population through growth and scrappage
2a	*Age 0*	
2b	*Age 1 through 29*	
2c	*Age 30 and higher*	
TAG-3	Calculate Analysis Year Travel Fraction	Calculation travel fraction across domain as function of age mix and annual miles traveled, by use type and age
3a	*Calculate total population within HPMS vehicle class*	
3b	*Calculate fraction within HPMS vehicle class*	
3c	*Compute travel fraction*	
TAG-4	Calculate Analysis Year VMT	If analysis year is in future, establish analysis year aggregate VMT across domain from base year VMT and growth
TAG-5	Allocate Analysis Year VMT By Roadway Type, Use Type, Age	Allocate aggregate VMT to roadway type, use type and age
TAG-6	Temporally Allocate Annual VMT to Hour by Roadway Type, Use Type, Age	Allocate annual VMT to hourly VMT
TAG-7	Convert to Total Activity Basis By Process	Convert to total activity basis at domain level: SHO, Starts, Extended Idle Hours
7a	*Calculate average speed*	
7b	*Convert to SHO*	

7c	*Convert SHO to Starts*	
7d	*Convert SHO to Exetended Idle Hours*	
TAG-8	Allocate Total Activity Basis By Zone Location	Allocate total activity to zone locations using geographic allocation factors
8a	*Allocate SHO to Zones*	
8b	*Allocate Starts to Zones*	
8c	*Allocate Extended Idle Hours to Zones*	
TAG-9	Calculate Distance Traveled	Calculate SHO.distance from SHO and average speeds

Some overall considerations when performing these calculations are:

1. The TotalActivityGenerator signs up for the Master Loop at the Year level which means the calculations are performed individually for each year at each location (i.e. link) for each emission process requiring SHO, start, and extended idle hour activity information.

2. The MOVES design allows the user to provide some or all of the values in core model input tables such as SHO, Starts and ExtendedIdleHours. The InputDataManager places any such user-supplied values in the MOVESExecution database before the TotalActivityGenerator is activated. The TotalActivityGenerator must not replace such user-supplied values.

3. When the Total Activity Generator encounters a missing value when performing a calculation, the result of the calculation is considered as missing. Records for which the results are missing are left out of the database and are not represented by a value of zero or some other numeric value or character.

Detailed descriptions of the calculations in each TAG follow. Each of the variables used in the TAG calculations either exists in the MOVESExecution database or is calculated by a previous TAG step. All of the MOVES2004 variables are described in the database documentation included in the MOVES database. The table in which each variable can be found is indicated in parentheses.

10.5.1. TAG-0: Determine the Base Year

Before any calculations can be done, the appropriate base year must be determined using the year of analysis and the isBaseYear information in the Year table.

Input Variables:

IsBaseYear (Year)
YearID (Key)

Output Variable:

BaseYear

Calculation:

If IsBaseYear(YearID) = "Y", then BaseYear=YearID.

Else

Find the maximum value of Year which is less than the current YearID for which IsBaseYearID(Year)= "Y", then BaseYear=Year.

10.5.2. TAG-1: Calculate Base Year Vehicle Population By Age.

Input Variables:

SourceTypePopulation (SourceTypeYear)
AgeFraction (SourceTypeAgeDistribution)

Output Variable:

SourceTypeAgePopulation

Calculation:

SourceTypeAgePopulation (YearID, SourceTypeID, AgeID) =
SourceTypePopulation (YearID, SourceTypeID) *
AgeFraction (YearID, SourceTypeID, AgeID)

10.5.3. TAG-2: Grow Vehicle Population from Base Year to Analysis Year

NOTE: This step is only required if the analysis year is in the future relative to the base year. For future projection, these TAG-2 steps are repeated in every year until the analysis year is reached.

TAG-2a: Age Zero (New Vehicles)

This calculation applies only to AgeID=0.

Input Variables:

SourceTypeAgePopulation (for AgeID=0 and YearID=YearID-1)
SalesGrowthFactor (SourceTypeYear)
MigrationRate (for YearID and YearID-1) (SourceTypeYear)

Output Variable:

SourceTypeAgePopulation (for AgeID=0 in current YearID)

Calculation:

SourceTypeAgePopulation (YearID, SourceTypeID, AgeID=0) =

[SourceTypeAgePopulation (YearID-1, SourceTypeID, AgeID=0) /
MigrationRate (YearID-1, SourceTypeID)] *
SalesGrowthFactor (YearID, SourceTypeID) *
MigrationRate (YearID, SourceTypeID)

Note: the full equation would divide through by age=0 survival rate to derive new sales in previous years prior to scrappage, and multiply by the same term to account for scrappage in the first year. Since scrappage is the same in all years, they cancel each other out, and these terms are excluded from the equation.

TAG-2b: Ages 1 through 29

This calculation loops through AgeID=x, where x is 1 through 29.

Input Variables:

SourceTypeAgePopulation (for AgeID=x and YearID=YearID-1)
SurvivalRate (for AgeID=x) (SourceTypeAge)
MigrationRate (SourceTypeYear)

Output Variable:

SourceTypeAgePopulation (for AgeID=1 through 29 in current YearID)

Calculation:

SourceTypeAgePopulation (YearID, SourceTypeID, AgeID=x) =
SourceTypeAgePopulation (YearID-1, SourceTypeID, AgeID=x-1) *
SurvivalRate (SourceTypeID, AgeID=x-1) *
MigrationRate (YearID, SourceTypeID)

TAG-2c: Age 30

This calculation is for Age 30, which includes years 30 and higher.

Input Variables:

SourceTypeAgePopulation (for AgeID=29 & 30 and YearID=YearID-1)
SurvivalRate (for AgeID=29 & 30) (SourceTypeAge)
MigrationRate (SourceTypeYear)

Output Variable:

SourceTypeAgePopulation (for AgeID=30 in current YearID)

Calculation:

SourceTypeAgePopulation (YearID, SourceTypeID, AgeID=30) =
SourceTypeAgePopulation (YearID-1, SourceTypeID, AgeID=29) *
SurvivalRate (SourceTypeID, AgeID=29) *
MigrationRate (YearID, SourceTypeID) +
SourceTypeAgePopulation (YearID-1, SourceTypeID, AgeID=30) *
SurvivalRate (SourceTypeID, AgeID=30) *
MigrationRate (YearID, SourceTypeID)

10.5.4. TAG-3: Calculate Analysis Year Travel Fraction

TAG-3a: Calculate total population within Highway Performance Management System (HPMS) vehicle class.

Input Variables:

SourceTypePopulation (for each SourceTypeID within HPMSVtypeID) (Source TypeYear)

HPMSVtypeID (HPMSVtype)

Output Variable:

HPMSVtypePopulation

Calculation:

HPMSVTypePopulation (YearID, HPMSVtypeID) =
Sum of [SourceTypeAgePopulation (YearID, SourceTypeID, AgeID)]
over all AgeID,
for all SourceTypeID within each HPMSVtypeID.

TAG-3b: Calculate fraction within HPMS vehicle class

Input Variables:

SourceTypeAgePopulation
HPMSVtypePopulation

Output Variable:

FractionwithinHPMSVtype

Calculation:

FractionwithinHPMSVtype (YearID, SourceTypeID, AgeID) =
SourceTypeAgePopulation (YearID, SourceTypeID, AgeID) /
HPMSVTypePopulation (YearID, HPMSVtypeID, SourceTypeID)

TAG-3c: Compute travel fraction

Input Variables:

FractionwithinHPMSVtype
RelativeMAR (SourceTypeAge)

Output Variable:

TravelFraction

Calculation:

TravelFraction (YearID, SourceTypeID, AgeID) =
(FractionwithinHPMSVtype (YearID, SourceTypeID, AgeID=x) *
RelativeMAR (SourceTypeID, AgeID=x)) /
Sum of [FractionwithinHPMSVtype (YearID, SourceTypeID, AgeID=x) *
RelativeMAR (SourceTypeID, AgeID=x)] over all AgeID and for all
SourceTypeID within each HPMSVtype.

10.5.5. TAG-4: Calculate Analysis Year VMT

Determine the total VMT within each HPMS vehicle type, accounting for VMT growth. This calculation is repeated in every YearID until the analysis year is reached.

TAG-4a: Base year.

In the base year, the AnalysisYearVMT is the same as the HPMSBaseYearVMT.

Input Variables:

HPMSBaseYearVMT (YearID=BaseYear) (HPMSVtypeYear)

Output Variable:

AnalysisYearVMT

Calculation:

AnalysisYearVMT (YearID, HPMSVTypeID) =
HPMSBaseYearVMT (YearID, HPMSVTypeID)

TAG-4b: All years following the base year.

In the any years following the base year, the AnalysisYearVMT is calculated from the previous year's value for AnalysisYearVMT.

Input Variables:

AnalysisYearVMT (YearID-1)
VMTGrowthFactor (HPMSVtypeYear)

Output Variable:

AnalysisYearVMT

Calculation:

AnalysisYearVMT (YearID, HPMSVTypeID) =
AnalysisYearVMT (YearID-1, HPMSVTypeID) *
VMTGrowthFactor (YearID, HPMSVTypeID)

10.5.6. TAG-5: Allocate Analysis Year VMT by Roadway Type, Use Type, Age

Input Variables:

AnalysisYearVMT
RoadTypeVMTFraction (RoadTypeDistribution)
TravelFraction

Output Variable:

AnnualVMTbyAgeRoadway

Calculation:

AnnualVMTbyAgeRoadway (YearID, RoadTypeID, SourceTypeID, AgeID) =
AnalysisYearVMT (YearID, HPMSVTypeID) *
RoadTypeVMTFraction (RoadTypeID, SourceTypeID) *
TravelFraction (YearID, SourceTypeID, AgeID)

10.5.7. TAG-6: Temporally Allocate Annual VMT to Hour

Input Variables:

AnnualVMTbyAgeRoadway
MonthVMTFraction (MonthVMTFraction)
DayVMTFraction (DayVMTFraction)
HourVMTFraction (HourVMTFraction)

Output Variable:

VMTbyAgeRoadwayHour

Calculation:

VMTbyAgeRoadwayHour (YearID, RoadTypeID, SourceTypeID, AgeID,
MonthID, DayID, HourID) =
AnnualVMTbyAgeRoadway (YearID, RoadTypeID, SourceTypeID, AgeID) *
MonthVMTFraction (SourceTypeID, IsLeapYear, MonthID) *
DayVMTFraction (RoadTypeID, SourceTypeID, MonthID, DayID) *
HourVMTFraction (RoadTypeID, SourceTypeID, DayID, HourID)
* Number of Days in MonthID/7

10.5.8. TAG-7: Convert to Total Activity Basis

Tag-7a: Compute average speed by roadway type

Input Variables:

AvgBinSpeed (AvgSpeedBin)
AverageSpeedFraction (AverageSpeedDistribution)

Output Variable:

AverageSpeed

Calculation:

AverageSpeed (RoadTypeID, SourceTypeID, DayID, HourID) =
Sum of [AvgBinSpeed(AvgSpeedBin) *
AverageSpeedFraction(RoadTypeID, SourceTypeID, DayID, HourID,
AvgSpeedBin)] over all AvgSpeedBin

Tag-7b: Convert VMT to SHO

Input Variables:

VMTbyAgeRoadwayHour
AverageSpeed

Output Variable:

SHObyAgeRoadwayHour

Calculation:

SHObyAgeRoadwayHour (YearID, RoadtypeID, SourceTypeID, AgeID,
MonthID, DayID, HourID) =
VMTbyAgeRoadwayHour(YearID, RoadtypeID, SourceTypeID, AgeID,
MonthID, DayID, HourID) *
AverageSpeed (RoadtypeID, SourceTypeID, DayID, HourID)

Tag-7c: Convert SHO to Starts

Input Variables:

SHObyAgeRoadwayHour
startsPerSHO (SourceTypeHour)

Output Variable:

StartsByAgeHour

Calculation:

StartsByAgeHour (YearID, SourceTypeID, AgeID, MonthID, DayID, HourID)
=
(Sum over all Roadtypes (SHObyAgeRoadwayHour(YearID, RoadtypeID,
SourceTypeID, AgeID, MonthID, DayID, HourID))) *
StartsPerSHO(SourceType, HourID, DayID)

Tag-7d: Convert SHO to Extended Idle Hours

Input Variables:

SHObyAgeRoadwayHour
idleSHOFactor (SourceTypeHour)

Output Variable:

idleHoursbyAgeHour

Calculation:

idleHoursbyAgeHour (YearID, SourceTypeID, AgeID, MonthID, DayID, HourID) =
(Sum over 24 hours and over all Roadtypes (SHObyAgeRoadwayHour(YearID, RoadtypeID, SourceTypeID, AgeID, MonthID, DayID, HourID))) *
idleSHOFactor(SourceType, DayID, HourID)

10.5.9. TAG-8: Allocate Total Activity to Zone Location

Tag-8a: Allocate SHO to Zones

Input Variables:

SHObyZoneAgeRoadwayHour
SHOAllocationFactor (ZoneYearRoadwayType)

Output Variable:

SHO (SHO)
This result populates the SHO field in SHO table.

Calculation:

SHO(YearID, ZoneID, RoadtypeID, SourceTypeID, AgeID, MonthID, DayID, HourID) =
SHObyAgeRoadwayHour(YearID, RoadtypeID, SourceTypeID, AgeID, MonthID, DayID, HourID) *
SHOAllocationFactor(YearID, ZoneID, RoadtypeID)

Tag-8b: Allocate Starts to Zones

Input Variables:

StartsByAgeHour
startAllocFactor (ZoneYear)

Output Variable:

Starts
Result of TAG-8b populates "starts" field of Starts Table

Calculation:

Starts(YearID, ZoneID, SourceTypeID, AgeID, MonthID, DayID, HourID) =
StartsByAgeHour(YearID, SourceTypeID, AgeID, MonthID, DayID, HourID) *
startAllocFactor(YearID, ZoneID)

Tag-8c: Allocate Extended Idle Hours to Zones

Input Variables:

idleHoursbyAgeHour
idleAllocFactor (ZoneYear)

Output Variable:

extendedIdleHours
Result of TAG-8c populates "extendedIdleHours" field of ExtendedIdleHours Table

Calculation:

extendedIdleHours(YearID, ZoneID, SourceTypeID, AgeID, MonthID, DayID, HourID) =
idleHoursbyAgeHour(YearID, SourceTypeID, AgeID, MonthID, DayID, HourID) * idleAllocFactor(YearID, ZoneID)

10.5.10. TAG-9: Calculate Distance Traveled Corresponding to Source Hours Operating

If the “Distance Traveled” output is requested by the RunSpec, (which also implies that some pollutant has been selected for the Running process), the distance field in the SHO table is calculated. Otherwise this distance field is output by this generator as NULL. The method used to produce this distance information involves multiplying the number of source hours operating (SHO) by the average vehicle speed associated with them. These average speeds have been calculated in Step 7a.

Input Variables:

SHO from step TAG-8a
Average Speed from step TAG7-A

Output Variable:

Distance (in SHO table)

Calculation:

Distance (YearID, MonthID, LinkID (ZoneID with RoadTypeID),HourID, DayID, AgeID, SourceTypeID) =
SHO((YearID, MonthID, LinkID (ZoneID with RoadTypeID),HourID, DayID, AgeID, SourceTypeID) * AverageSpeed(RoadTypeID, SourceTypeID, DayID, HourID)

10.6. OperatingModeDistributionGenerator (OMDG)

The OperatingModeDistributionGenerator is only relevant to the total energy – running pollutant-process. This is the only combination of pollutant and process which has multiple operating modes in MOVES2004. For this pollutant and process the OMDG calculates the distribution of operating modes for each source type on each roadway type modeled using data from the MOVESExecution database. The resulting distributions are added to the OperatingModeDistribution core model input table.

The method used to generate these operating mode distributions in MOVES2004 is a refinement of the method described in section 7.1.3 of the *Draft Design and Implementation Plan for MOVES*. The task of the OMDG is to produce operating mode distributions, in terms of a set of VSP-and-speed-range-based operating modes, for each combination of source type, road type, hour of the day, and day of the week using as input average speed distribution information for each such combination. Driving schedules representing typical operation at different average speeds for each source type operating on each road type play an intermediate role in translating average speed information into VSP distributions. Each average speed bin used in the input average speed distributions is represented by a pair of “bracketing” driving schedules one of which has a slightly higher average speed and one of which has a slightly lower average speed. VSP is calculated on a second by second basis for the source use type operating over these two schedules and the results are weighted appropriately to represent the average speed distribution. A full discussion of the operating mode definitions and the use of vehicle specific power (VSP) and driving schedules in MOVES2004 is contained in a separate report, *MOVES2004 Energy and Emissions Inputs*, downloadable from the MOVES web site.

This algorithm is divided into seven steps referred to as OMDGs (operating mode distribution generator steps) in this document. Most OMDGs reference the MOVESExecution database and all implement a simple mathematical formula. Steps 1-3 take snippets of actual on-road driving schedules stored in the MOVESExecution database, and determine the mix of these schedules to use based on the mix of average speeds and roadway types; Step 4 calculates VSP, Step 5 determines the operating mode bin (based on speed and VSP), and Steps 6 and 7 determine the overall mix of operating mode bins based on the mix of roadway types and average speed.

A brief description of each of the eight OMDGs is shown in table 10-4:

Table 10-4. Overview of Calculation Steps

Step	**Description**
OMDG-1	Determine bracketing drive schedules
OMDG-2	Determine proportions for bracketing drive schedules
OMDG-3	Determine distribution of drive schedules, including a sub-step to reflect separation of ramp and non-ramp freeway driving
OMDG-4	Calculate second-by-second vehicle specific power (VSP)
OMDG-5	Determine operating mode bin for each second
OMDG-6	Calculate operating mode fractions for each drive schedule
OMDG-7	Calculate overall operating mode fractions

The Operating Mode Distribution Generator signs up for the Master Loop at the Year level which means that it executes for each year in the run specification for each Link location for the running emission process.

The MOVES design allows the user to directly provide some or all of the operating mode distribution values in core model input tables such as OpModeDistribution. The InputDataManager places these user-supplied values in the MOVESExecution database OpModeDistribution table before the Operating Mode Distribution Generator is activated. The Operating Mode Distribution Generator does not replace any such user supplied values.

When the Operating Mode Distribution Generator encounters a missing value when making a calculation, the results of the calculation are considered as missing. Such missing results are left out of the database and are not represented by a value of zero or some other numeric value or character.

The detailed descriptions of the calculations in each OMDG step are as follows: Each of the variables used in the OMDG calculations either exists in the MOVESExecution database or is calculated by a previous OMDG step. All of the MOVESExecution variables are described in the database documentation.

10.6.1. OMDG-1: Determine bracketing drive schedules

Each average speed bin lies between (is bracketed) by the average speeds of two drive schedules. This step determines which two drive schedules bracket the average speed bin and stores the identity and average speeds of the two bins. This is done for each source type, and roadway type for each average speed bin.

Input Variables:

AvgBinSpeed (AvgSpeedBinID) from the AvgSpeedBin table.
AverageSpeed(DriveScheduleID) from the DriveSchedule table.

Output Variables:

BracketScheduleLo (SourceTypeID, RoadTypeID, AvgSpeedBinID)
LoScheduleSpeed (SourceTypeID, RoadTypeID, AvgSpeedBinID)
BracketScheduleHi (SourceTypeID, RoadTypeID, AvgSpeedBinID)
HiScheduleSpeed (SourceTypeID, RoadTypeID, AvgSpeedBinID)

Calculation:

For each SourceTypeID, RoadTypeID, and AvgSpeedBinID:

The output variables are determined using table DriveSchedule, where for each combination of SourceTypeID, RoadTypeID, AvgSpeedBinID, BracketScheduleLo and BracketScheduleHi are determined as the drive schedules associated with the source type and road type with the next lowest and next highest average speeds compared to the value of AvgBinSpeed. LoScheduleSpeed and HiScheduleSpeed are the values of AverageSpeed for the bracket schedules. The DriveScheduleAssoc table contains the information as to which DriveScheduleID's are associated with which SourceTypeID's and which RoadTypeID's.

10.6.2. OMDG-2: Determine proportions for bracketing drive schedules

This step determines the proportion of each of the bracketing drive schedules such that the combination of the average speeds of drive schedules equals the nominal average speed of each average speed bin. The results are then weighted by the fraction of all operating time that are represented by the time spent in that average speed bin. This is done for each source type, roadway type, day of week and hour of day.

Input Variables:

AvgBinSpeed (AvgSpeedBinID) from the AvgSpeedBin table.
AvgSpeedFraction (SourceTypeID, RoadTypeID, HourDayID, AvgSpeedBinID) from the AvgSpeedDistribution table.

with

BracketScheduleLo (SourceTypeID, RoadTypeID, AvgSpeedBinID)
LoScheduleSpeed (SourceTypeID, RoadTypeID, AvgSpeedBinID)
BracketScheduleHi (SourceTypeID, RoadTypeID, AvgSpeedBinID)
HiScheduleSpeed (SourceTypeID, RoadTypeID, AvgSpeedBinID)

Output Variable:

LoScheduleFraction (SourceTypeID, RoadTypeID, HourDayID, AvgSpeedBinID)
HiScheduleFraction (SourceTypeID, RoadTypeID, HourDayID, AvgSpeedBinID)

Calculation:

For each SourceTypeID, RoadTypeID, HourDayID and AvgSpeedBinID:
LoScheduleFraction =
(HiScheduleSpeed – AvgBinSpeed) /
(HiScheduleSpeed – LoScheduleSpeed)
If BracketScheduleHi=BracketScheduleLo (meaning the average speeds of the “Lo and “Hi” schedule is the same):
    LoScheduleFraction = 1
    HiScheduleFraction = (1 – LoScheduleFraction)
Weight the results by the fraction of all speeds that are represented by that average speed bin.
LoScheduleFraction = LoScheduleFraction * AvgSpeedFraction
HiScheduleFraction = HiScheduleFraction * AvgSpeedFraction

10.6.3. OMDG-3: Determine distribution of drive schedules

This step includes a preliminary sub-step which accounts for the effect of ramp driving. The rampFraction field in the RoadType table indicates the fraction of time on each roadway type which is spent on ramps. The “isRamp” field in the DriveScheduleAssoc table determines whether a schedule is to be associated with ramp driving or with driving on roadways exclusive of ramps. The MOVESDefault database provided with MOVES2004 considers that Ramp driving occurs on freeways and interstates. Other roadway types have no ramp activity (zero ramp fraction).

User attempting to modify the database in this area should keep in mind several data constraints: There must always be <u>at least one</u> driving schedule not indicated as a ramp which is associated with each combination of source type and roadway type in the DriveScheduleAssoc table. Then additionally, for every ramp fraction which is greater than zero in the RoadType table, there must be <u>exactly one</u> driving cycle which is indicated as a ramp for each source type using that roadtype. (Conversely Ramp fractions can be zero even if there is an associated ramp driving schedule; in this case the ramp schedule is simply not used.)

OMDG-3a: Determine distribution of ramp schedules

This step is accounts for freeway and interstate ramps, based on the input field RampFraction, contained in table RoadType. For any driving schedule which is associated with the source type and roadway type which has an IsRamp value of "Y", the ramp fraction is added into DriveScheduleFraction for that driving schedule. This is done for each source type, roadway type, day of week and hour of day.

Input Variables:

RampFraction(RoadTypeID)
DriveScheduleID (SourceTypeID, RoadTypeID, IsRamp=Y)

Output Variable:

DriveScheduleFraction (SourceTypeID, RoadTypeID, HourDayID, DriveScheduleID)

Calculation:

DriveScheduleFraction = RampFraction(RoadTypeID)

OMDG-3b: Determine distribution of non-ramp schedules

This calculation adjusts for the RampFraction on the given roadway. This step determines the distribution of drive schedules which represents the sum of all of the average speed bins. This is done for each source type, roadway type, day of week and hour of day for all driving schedules which are associated with the source type and roadway type which have an IsRamp value of "N".

Input Variables:

LoScheduleFraction (SourceTypeID, RoadTypeID, HourDayID, AvgSpeedBinID)
HiScheduleFraction (SourceTypeID, RoadTypeID, HourDayID, AvgSpeedBinID)
RampFraction(RoadTypeID)
DriveScheduleID (SourceTypeID, RoadTypeID, IsRamp=N)

Output Variable:

DriveScheduleFraction (SourceTypeID, RoadTypeID, HourDayID, DriveScheduleID)

Calculation:

For each SourceTypeID, RoadTypeID, HourDayID and DriveScheduleID:
DriveScheduleFraction =
[(Sum Of LoScheduleFraction over all cases where BracketScheduleLo = DriveScheduleID) + (Sum Of HiScheduleFraction over all cases where BracketScheduleHi =
DriveScheduleID)] * (1-RampFraction)

10.6.4. OMDG-4: Calculate second-by-second vehicle specific power (VSP)

This step calculates the vehicle specific power (VSP) for each drive schedule for each source type. This is done for each source type, drive schedule and second.

Input Variables:

Speed (DriveScheduleID, Second) from the DriveScheduleSecond table.
RollingTermA (SourceTypeID) from the SourceUseType table.
RotatingTermB (SourceTypeID) from the SourceUseType table.
DragTermC (SourceTypeID) from the SourceUseType table.
SourceMass (SourceTypeID) from the SourceUseType table.

Output Variable:

VSP (SourceTypeID, DriveScheduleID, Second)
Accel (SourceTypeID, DriveScheduleID, Second)

Calculation:

For each SourceTypeID, DriveScheduleID and Second:
Preliminary Calculations:
Speed (meters/second) = Speed (mph) * 0.447 m/s per mph.
Accel (meters/second2)= Speed (t) – Speed (t-1), with Speed in meters/second.
VSP =
[RollingTermA*Speed
+ RotatingTermB * Speed^2
+ DragTermC * Speed^3
+ SourceMass * Speed * Accel]
/ SourceMass
(Speed in meters/second, Accel in meters/second^2)

10.6.5. OMDG-5: Determine operating mode bin for each second

This step accounts for VSP, speed and accel. The VSP value for each second is compared to the upper and lower bounds for the operating mode bins and a bin ID is assigned to each second. This is done for each source type, drive schedule and second.

Input Variables:

VSPLower(OpModeID) from the OperatingMode table.
VSPUpper(OpModeID) from the OperatingMode table.
SpeedLower(OpModeID) from the OperatingMode table.
SpeedUpper(OpModeID) from the OperatingMode table.
BrakeRate1Sec(OpModeID) from the OperatingMode table.
BrakeRate3Sec(OpModeID) from the OperatingMode table.
Speed (SourceTypeID, DriveScheduleID, Second) from DriveScheduleSecond with

VSP (SourceTypeID, DriveScheduleID, Second) from OMDG-4
Accel (SourceTypeID, DriveScheduleID, Second) from OMDG-4

Output Variable:

OpModeIDbySecond (SourceTypeID, DriveScheduleID, Second)

Calculation:

For each SourceTypeID, DriveScheduleID and Second:
An OpModeID is assigned for each second in each drive schedule based on where the VSP, Speed and Accel values in that second falls. VSP and speed are compared against the VSP and speed bounds to determine to appropriate bins. Then the braking (ID=0) and idle bins (ID=1) should be assigned based on Accel and Speed, respectively. This sequence is important since there is overlap in the definitions between the non-braking/idle bins and the braking/idle bins.

10.6.6. OMDG-6: Calculate operating mode fractions for each drive schedule

Once all the seconds in each operating mode bin is known, the distribution of the bins can be determined. The sum of the operating mode fractions sums to one for each source type and drive schedule combination. This is done for each source type and drive schedule.

Input Variables:

OpModeIDbySecond (SourceTypeID, DriveScheduleID, Second)

Output Variable:

OpModeFractionbySchedule (SourceTypeID, DriveScheduleID, OpModeID)

Calculation:

For each SourceTypeID, DriveScheduleID and OpModeID:

OpModeFractionbySchedule =
(Number of Seconds in OpModeID during DriveSchedule) /
(Total Number of Seconds in DriveSchedule)

10.6.7. OMDG-7: Calculate overall operating mode fractions

This step calculates the overall operating mode fractions by weighting the operating mode fractions of each drive schedule by the drive schedule fractions. This is done for each source type, road type, day of the week, hour of the day and operating mode.

Input Variables:

OpModeFractionbySchedule (SourceTypeID, DriveScheduleID, OpModeID)
DriveScheduleFraction (SourceTypeID, RoadTypeID, HourDayID, DriveScheduleID)

Output Variable:

OpModeFraction (SourceTypeID, RoadTypeID, HourDayID, PolProcessID, OpModeID)

Calculation:

The operating mode distribution generator developed in this work assignment is for only the running process of the total energy pollutant.
For each SourceTypeID, RoadTypeID, HourDayID and OpModeID:
OpModeFraction =
Sum of OpModeFractionbySchedule*DriveScheduleFraction over all DriveSchedules

The Results of OMDG-7 populate the OpModeDistribution table of the Execution Location Database.

The OpModeFraction in the OpModeDistribution table is:

OpModeFraction (SourceTypeID, LinkID, HourDayID, PolProcessID, OpModeID)

The value of LinkID needed for this table is determined from RoadTypeID and ZoneID.

10.7. Source Bin Distribution Generator (SBDG)

The Source Bin Distribution Generator produces the distribution of source bins by source type and model year. This information provides the mapping between the activity elements of MOVES (total activity and operating modes), which are based on source use type, and the emission rates, which are based on source bin. The SBDG takes as input fleet distributions of source bin categories (e.g. weight class, engine size, fuel type etc.) by model year.

Data about the characteristics of the existing and projected vehicle fleet are stored in several of the tables within the MOVESExecution database as shown in table 10-5. The Source Bin Distribution Generator uses information in the first seven of these tables to populate the last two: SourceBin and SourceBinDistribution which are core model input tables.

Table 10-5. Tables used by SourceBinGenerator

Table Name	**Key Fields**	**Additional Fields**	**Notes**
SourceTypePolProcess	sourceTypeID polProcessID	isSizeWeightReqd isRegClassReqd isMYGroupReqd	Indicates which pollutant-processes the source bin distributions may be applied to and indicates which discriminators are relevant for each sourceType and polProcess
FuelEngFraction	sourceTypeModelYearID fuelTypeID engTechID	fuelEngFraction	Joint distribution of vehicles with a given fuel type and engine technology. Sums to one for each sourceType & modelYear
SizeWeightFraction	sourceTypeModelYearID fuelTypeID engTechID weightClassID engSizeID	sizeWeightFraction	Joint distribution of engine size and weight. Sums to one for each sourceType, modelYear and fuel/engtech combination.
RegClassFraction	sourceTypeModelYearID fuelTypeID engTechID regClassID	regClassFraction	Fraction of vehicles in a regulatory class.. Sums to one for each sourceType, modelYear and fuel/engtech combination.
PollutantProcessModel Year	PolProcessID modelYearID	modelYearGroupID	Defines model year groups.
ModelYearGroup	modelYearGroupID	shortModYrGroupID modelYearGroupName	Defines short model year group Ids.
SourceTypeModelYear	sourceTypeModelYearID	modelYearGroupID modelYearID sourceTypeID	Decodes ID field into modelYearID and sourceTypeID

SourceBin	SourceBinID	engSizeID fuelTypeID engTechID emisTechID modelYearGroupID weightClassID	List of sourceBins
SourceBinDistribution	SourceTypeModelYearID PolProcessID sourceBinID	sourceBinActivityFract ion	Stores source bin distributions.

The SourceBinGenerator uses information from the run specification to determine which sourcetypes, modelyears, fuel-types, pollutants and processes are relevant, and limits the Generator action to the relevant sources.

"Source bin discriminators" refer to characteristics that are used to distinguish the source bins. The MOVES source bin discriminators are:

fuelType*
modelYearGroup
engTech*
regClass
engSize*
weight Class*
(* fuelType and engTech are distinct but highly correlated, and, thus, handled together; similarly engSize and weightClass are handled together)

a. Because it would be awkward to have all these separate fields in each database table related to source bins, this information is combined into a single unique BIGINT identifier 1ffttrryysssswwww00, where the digits represent the bin discriminators as follows:

Leading 1 (1)
Fuel (ff)
EngineTech (tt)
Regulatory Class (rr)
Model Year Group (yy)
Engine Size (ssss)
Engine Weight (wwww)
Extra zeros (00)
With the exception of the ModelYear group, each bin discriminator is coded as for the discriminator ID (see the database attribute table), with leading zeros added as needed. ModelYear group is coded using the shortModYrGroupID as defined in the database table ModelYearGroup.

The ordering of the subfields within this identifier is essentially arbitrary. The SourceBin table contains both the individual fields and the combined source bin identifier and can be used to "decode" the combined identier values.

b. The SourceTypePolProcess table identifies which discriminators are relevant for each source type and pollutant/emission process (polProcess). If a discriminator is not relevant for a given source type & polProcess, the discriminator ID is set to 0 (which means that the discriminator value doesn't matter) and the fraction of vehicles to which this value of this discriminator applies is set to 1 for all model years.

c. The SourceBinGenerator "signs up" with the MOVES MasterLoop at the PolProcess level which means it executes only once for each relevant process (running exhaust, start exhaust, and extended idle exhaust).

d. For each RunSpec-required source type, polProcess, fuel type and model year (as implied by the selected year and the largest valid ageID), and for each relevant/existing combination of sourcebin discriminators (as determined by the input tables), the sourceBinActivityFraction is calculated as follows:

$$\text{sourceBinActivityFraction} = \text{fuelEngFraction} * \text{regClassFraction} * \text{sizeWeightFraction}$$

e. The SourceBinDistribution table lists the sourceBinActivityFractions for each sourceBin, sourceType, modelYear and polProcess. The set of such records for a given sourceType, modelyear, and polProcess constitute a source bin distribution which sums to unity.

f. The SourceBin table provides a list of unique source bins that have a SourceBinActivityFraction > 0 in at least one source bin distribution. The SBDG adds sourceBinID records to this table if necessary, but does not duplicate records already present.

g. If data is not available for all the model years implied by the selected year, source bin distributions for the missing years prior to the first populated for the same sourceType and polProcess are generated which are equal to the distribution from that first populated year. Source bin distributions are also generated for missing years after the last populated for the same sourceType and polProcess which are equal to the distribution from that last populated model year. Years extrapolated are only those earlier than the earliest model year for which data is provided or later than the latest model year for which data is provided.

h. If any data is already present in the CMIT SourceBinDistribution table for a combination of sourceType and polProcess, which would mean that the user had entered this information directly, this Generator does not produce SourceBinDistribution output for that combination.

The approach to performing the calculation steps implied by these specifications is shown in table 10-6:

Table 10-6. SourceBin Generator Calculation Steps

Step/Query	**Description**
1 populate sb_fractions	Select data from the SourcBinGenerator tables to fill a temporary table with identifying and distribution records for each relevant combination of sourceType, pollutant-process and bin identifier.
2_1 sb_fractions update regulatory class.	For each record, for pollutant-processes that do not require regulatory class information, set regClassID to 0 and regClass Fraction to 1.
2_2 sb_fractions update MY GroupID	For each record, for pollutant-processes that do not require model year group information, set MYGroupID to 0.
2_3 sb_fractions update size and weight	For each record, for pollutant-processes that do not require size and weight information, set engSizeID and weightClassID to 0 and sizeWeightFraction to 1.
2_5 calculate sb_fractions	For each record, calculate sb_fractions as the product of fuelEngFraction, emisTechFraction, and sizeWeightFraction.
2_7 populate sb_fractions no dups	Remove duplicate records from sb_fractions; save as a second temporary table, “sb_fractions no dups”.
4 populate SourceBin	Select unique sourceBinIDs with fractions >0 to fill SourceBin table.
4_5 generate bin IDs	Generate sourceBinIDs from bin discriminators.
5 append source bin distributions	Use SourceBin table and “sb_fractions no dups” to fill SourceBinDistribution table.

10.8. Meterology Generator

Basic meteorological parameters such as temperature, humidity and heat index are stored in the MOVESExecution database table ZoneMonthHour. This generator uses the temperature and relative humidity information in the ZoneMonthHour Table and performs the necessary calculations to populate the heatIndex field in this table.

Since the heatIndex field is used only in the calculation of air conditioning factors used to adjust the total energy consumption of the running process, this generator is instantiated only if the total energy consumption – running pollutant process is included in the run. It subscribes to the MOVES master loop mechanism at the process level for only the running process.

The Meteorological Generator uses the temperature and relative humidity data in the ZoneMonthHour table to populate the Heat Index column.

For each RunSpec-required zone, month and hour, the Heat Index is calculated by following the algorithm:

```
HI = -42.379 + 2.04901523*T + 10.14333127*RH + -0.22475541*T*RH +
     -6.83783 *0.001*T*T + -5.481717 * 0.01 * RH*RH +
     1.22874*0.001*T*T*RH + 8.5282*0.0001*T*RH*RH +
     -1.99*.000001*T*T*RH*RH
```

where HI is the heat index, T is temperature in degrees F, and RH is the relative humidity in percent. This formula is a recent heat index algorithm used by the National Weather Service.

10.9. Alternative Vehicle Fuels and Technologies (AVFT) Strategy

The Alternative Vehicle Fuels & Technologies Strategy is the first implementation of an internal control strategy. This control strategy allows the user to input penetration rates (i.e. sales fractions) by model for a broad range of advanced technologies, through model year 2050. It is thus a key element in the ability of MOVES to perform “what-if” analysis.

The AVFT has two components. One component is incorporated into the MOVES GUI and allows the user to create specifications for replacement inputs for the FuelEngFraction table and, indirectly, the SizeWeightFraction and RegClassFraction tables. A second portion, the actual InternalControlStrategy MasterLoopable object, uses these specifications to produce these replacement tables prior to the running of the SourceBinGenerator. This causes changes to the SourceBinDistributions and the eventual MOVES output.

The user has the option to save the specifications (AVFTspecs) for the replacement input tables. The specifications are saved as an XML file. Most of the content of these specifications is the data to fill a large FuelEngFraction table in the MOVES database. The user is able to save the AVFTSpecs and to load and modify a saved AVFTSpec. A RunSpec may have an associated AVFTSpec; but AVFTspecs may also exist independently of RunSpecs.

10.9.1. Functional Characteristics of the AVFT GUI Component:

a. "Strategies" appears on the RunSpec Navigation List immediately after "Manage Input Data Sets". "Alternative Vehicle Fuels & Technologies" is a sub-section. Control Strategies are not required, thus the initial status icon for Strategies (and of sub-sections) is the green check or the yellow squiggles depending on whether the Run Spec currently includes a control strategy that modifies the default data.

b. An AVFT file management panel appears when the user selects "Alternative Vehicle Fuels & Technologies" from the navigation list. The panel includes:

   The name of the associated AVFTSpec (if one exists), or an indicator that no AVFTSpec exists and that default Fuel/Technology fractions are being used. The name displayed changes in response to the buttons below. No more than one AVFTSpec may be associated with a given RunSpec.

   A "New" button to create a new AVFTSpec. (This has the same effect as returning to default and editing.)

   An "Edit" button to view & modify the associated AVFTSpec.

   An "Import" button to load/associate an existing AVFTSpec. This allows the user to browse for exported AVFTSpecs and to select one.

   An "Export" button allows the user to save a AVFTSpec independent of the RunSpec. (Note: Saving the RunSpec itself includes saving the associated AVFTSpec. Executing the RunSpec after editing an AVFTSpec but without saving includes execution of the new AVFTSpec.)

   A "Delete" button to remove an associated AVFTSpec and revert to MOVES defaults.

   A "Cancel" button to close the panel without changes.

c. An AVFT Edit panel appears next to the AVFT file management panel when the user selects "Edit" or "New" in that panel. The Edit panel contains a description button and a details subpanel.

   Description Button: The user may (optionally) enter a short text description of the spec.

   Details Panel:

   1. The user may select a SourceType from a drop down list of all SourceTypes. The panel then displays a table of the Fuel/Engine Technology fractions from the associated AVFTSpec (or MOVES default DB FuelEngFraction table) for that SourceType for model years 2001-and-later. (The default FuelEngFraction table need not include all model years and the AVFTSpec may include similarly limited model years. The panel shows the model years provided and allows users to add additional years if desired (see below). In any case, the MOVES SourceBinGenerator extrapolates needed future years by repeating values for the last year provided.)

   2. Fuel/EngineTechnologies are listed as columns. Model years are listed as rows. The first value in each row is the model year. Fractions are sorted into categories as listed in the in the FuelEngTechAssoc table for that SourceType. The categories are displayed in the order indicated by the CategoryDisplayOrder field of the FuelEngTechAssoc table. Within a category, fractions are listed in ascending order by FuelTypeID*100 +EngTechID. The final value in each row is the sum of all FuelEngFractions for that source/type model year (so the user can see if the values sum to 1 as desired.) This panel normally fills the entire screen, with scroll bars to display any portions that don't fit.

   3. The initial view displays fractions aggregated in the categories listed in the FuelEngTechAssoc table for that SourceType. The user can display or hide detailed views that display the fractions for individual Fuel/EngineTechnology combinations. Categories with only one member are not considered "aggregate" (ie, they may be modified in all views and there is no "detailed view" to display or hide). Columns of aggregate fractions should be labelled with the category

name. Columns of non-aggregrate fractions should be labeled with the name of the FuelType and the Engine Technology.

4. The user may select an "Add Model Years" button that adds one or more model year rows at the bottom. The user has an option to specify how many rows to add with this feature but the sytem currently will not add model years beyond 2050. In each new row, the model year field is the next consecutive model year. The initial values for the FuelEngFractions in the new row equal those of the previous model year.
5. The user may replace any (non-aggregate) fraction in the table. Category aggregate fractions are grayed-out to indicate that they must be modified in a detail view. Fractions must be between 0 and 1, inclusive. The GUI does not allow the user to enter values <0 or >1.
6. The user may manually assure that the fractions sum to 1 for a sourcetype/model year or may choose a "Normalize" button that will keep the proportions of the input fractions but adjust so they sum to 1. The AVFTSpec does not "Export" and the RunSpec does not "Save" or "Execute" until all sourcetypes/model years sum to 1. Error messages are provided to the user as needed.
7. If the FuelEngTechAssoc table lists only one FuelType/EngineTechnology combination for the SourceType (currently true for motorcycles), there is no action for the user to take. The GUI displays the appropriate detail screen (which should have only one Fuel/EngineTechnology column), and displays a message: "Only one FuelType/EngineTechnology combination is supported for (SourceType Name). No alternate values allowed."

10.9.2. Functional Characteristics of the AVFT MasterLoopable

a. The AVFT processor is MasterLoopable; executing after the InputDataManager and before the Generators. It is instantiated under the same conditions as the SourceBinGenerator and executes at the emission process master loop level.

b. The AVFT FuelEngFractions are processed to replace the default fractions in FuelEngFractions in the Execution Location Database. In particular, if the specification provides inputs for a given SourceType/ModelYear, those inputs replace all values for that SourceType/ModelYear. Empty records in both the original and replacement tables (ie, records for which the implied fuelEngFraction =0) are taken into account.

c. The default RegClassFractions and SizeWeightFractions are modified such that the aggregate (across FuelType/EngineTechnology) RegClassFractions for the SourceType/ModelYear remain the same after application of the AVFT. The new RegClassFractions and SizeWeightFractions conserve the original fractions for vehicles with unchanged FuelEng characteristics and proportionally distribute the remaining fractions among the "changed vehicles".

   This calculation is specified in detail below for RegClassFraction. The algorithm for SizeWeightFractions is analogous, except the individual Regulatory Classes are replaced by combinations of EngSizeID and WeightClassID.

Compute New Reg Class Fractions

For each source type and model year, and for each supported FuelEngTech combination (i), either there exists an original **FuelEngFraction(i)** and a **NewFuelEngFraction(i)**, or it is implied that the FuelEngFraction(i) or NewFuelEngFraction(i) =0.

```
Compute DeltaFuelEngFraction(i) as NewFuelEngFraction(i)-
FuelEngFraction(i).

Compute ProportionOfNew(i) as
    IF  DeltaFuelEngFraction(i)>=0
        Proportion =0
    ELSE
        ProportionOfNew =
            DeltaFuelEngFraction(i) /
            Sum (over all i where DeltaFuelEngFraction <0)
            (DeltaFuelEngFraction(i)).

Compute PortionOld(i) as
    IF NewFuelEngFraction(i)=0,
        PortionOld(i) =0
    ELSE PortionOld = Min (1,
    FuelEngFraction(i)/NewFuelEngFraction(i))
```

For each SourceType & Model Year there is a set of original RegClassFractions(i,j), where “i” indicates the associated FuelEng combination and “j” indicates the RegClass. These are used with the results of the above calculations to create NewRegClassFractions(i,j,).

```
For each j
    For each i
        Compute NewRegClassFraction(i,j) as:

            IF NewFuelEngFraction(i)= 0,
                NewRegClassFraction(i,j) =0
                (Doesn't need to be in database)
            ELSE
                NewRegClassFraction(i,j) =
                    PortionOld(i) * RegClassFraction (i,j) +
                    [1-PortionOld(i)) *
                    (Sum(over all i) (ProportionOfNew(i)
                    *RegClassFraction(i,j))]
```

10.10. Emission Calculators – Energy Consumption Calculator (ECC)

The energy consumption calculator calculates energy consumption (total, petroleum-based and fossil-based) for four processes: running, start, extended idle and well-to-pump, for each source type on each roadway type in MOVES2004. It uses input from CMITs produced by the total activity, operating mode distribution, source bin, and meteorology generators, and calculates the quantities of energy consumed in the form of the MOVES output database.

Note: This functional calculator is actaully implemented as two MOVES EmissionCalculator classes: an "EnergyConsumptionCalculator" for the running, start, and extended idle processes, and a "WellToPumpProcessor" which "chains" onto the EnergyConsumptionCalculator if theWellToPump process is involved.

10.10.1. Overview of Calculation Steps

Table 10-7. Overview of Emission Consumption Calculator

Step	
ECCP-1	Preliminary calculations
ECCP-1a	*Calculate petroleum and fossil energy fractions*
ECCP-1b	*Convert age to model year for analysis year*
ECCP-2	Calculate adjustments
ECCP-2a	*Calculate air conditioning adjustment*
ECCP-2b	*Calculate temperature adjustment*
ECCP-2c	*Calculate fuel adjustment*
ECCP-3	Calculate running energy consumption
ECCP-4	Calculate start energy consumption
ECCP-5	Calculate extended idle energy consumption
ECCP-6	Calculate well-to-pump energy consumption
ECCP-6a	*Calculate pump-to-wheel energy consumption*
ECCP-6b	*Interpolate well-to-pump factor by FuelSubType in analysis year*
ECCP-6c	*Calculate well-to-pump factor by FuelType in analysis year*
ECCP-6d	*Calculate well-to-pump energy consumption*

The level of aggregation for each calculation is dictated by the key fields of the input and output variables. The result of energy quantities in steps 2, 3, 4 and 5 are at the level dictated by the MOVES2004 output database design: calendar year, month, day, hour, county, zone, link, pollutant, process, source type, fuel type, model year, and road type.

If the EnergyConsumptionCalculator encounters a missing value when making a calculation, the result of the calculation is also considered to be missing. Such results are left out of the database and are not represented in the MOVES output by a value of zero or some other numeric value or character.

The EnergyConsumptionCalculator signs up for the Master Loop at the year level which means that it executes for each location (linked) for each calendar year. It signs up for the running, start and extended idle processes to the extent they are called for by the run specification.

10.10.2. Detailed Steps

ECCP-1: Preliminary Calculations

Step 1a: Calculate petroleum and fossil energy fractions by fuel type

Since petroleum and fossil energy fractions vary by fuel subtype, this step is required to aggregate these fractions up to the fuel type level.

Input Variables:

MarketShare (County, Year, MonthGroup, FuelSubType,)
From FuelSupply table
Note: If record not found in database, default value of 1.0 is used if fuelSubtypeID ends with 0, otherwise default value of 0.0 is used.
FuelSubtypePetroleumFraction (FuelSubType)
From FuelSubtype table
FuelSubtypeFossilFraction (FuelSubType)
From FuelSubtype table

Output Variable:

PetroleumFraction (County, Year, Month, FuelType)
FossilFraction (County, Year, Month, FuelType)

Calculation:

PetroleumFraction=

$$\sum_{n=1}^{no.ofFuelSubTypeswithinFuelType} \text{MarketShare} * \text{FuelSubtypePetroleumFraction}$$

FossilFraction =

$$\sum_{n=1}^{no.ofFuelSubTypeswithinFuelType} \text{MarketShare} * \text{FuelSubtypeFossilFraction}$$

Step 1b: Convert Age to Model Year For Analysis Year

Input Variables:

YearID
AgeID

Output Variable:

ModelYearID

Calculation:

ModelYearID = YearID - AgeID

ECCP-2: Calculate adjustments

Step 2a: Calculate air conditioning adjustment

Preliminary calculation (1): ACOnFraction

This step calculates the fraction of time the AC compressor is engaged, as a function of Heat Index

Input Variables:

HeatIndex (Zone, Month, Hour)
From ZoneMonthHour table
ACActivityTermA (MonthGroup, Hour)
From MonthGroupHour table
ACActivityTermB (MonthGroup, Hour)
From MonthGroupHour table
ACActivityTermC (MonthGroup, Hour)
From MonthGroupHour table

Calculation:

ACOnFraction (Zone, Month, Hour) =
(ACActivityTermA+ACActivityTermB*HeatIndex
+ACActivityTermC*HeatIndex^2)

If ACOnFraction<0, set to 0
If ACOnFraction>1, set to 1

Preliminary calculation (2): ACActivityFraction

This step calculates the overall A/C activity fraction, which accounts for the A/C on fraction, the penetration of A/C in the fleet and the fraction of those A/C systems that are functioning.

Input Variable

ACOnFraction (Zone, Month, Hour) from previous calculation

ACPenetrationFraction (SourceType, ModelYear)
From SourcetypeModelYear table
FunctioningACFraction (SourceType, Age)
From SourceTypeAge table

Calculation

ACActivityFraction (Zone, Month, Hour, SourceType, ModelYear) =
ACOnFraction
* ACPenetrationFraction
*FunctioningACFraction

Calculate AC Adjustment

Input Variables:

ACActivityFraction (Zone, Month, Hour, SourceType, ModelYear)
From previous calculation
FullACAdjustment (PolProcess, SourceType, OpMode)
From FullACAdjustment table

Calculation:

ACAdjustment (Zone, Month, Hour, SourceType, ModelYear, PolProcess, OpMode) = 1 + ((FullACAdjustment –1)*ACActivityAdjustment)

Step-2b: Calculate temperature adjustment

Input Variables:

Temperature (Zone, Month, Hour)
From ZoneMonthHour table
TempAdjustTermA (PolProcess, SourceType, FuelType)
From TemperatureAdjustment table
TempAdjustTermB (PolProcess, SourceType, FuelType)
From TemperatureAdjustment table
TempAdjustTermC (PolProcess, SourceType, FuelType)
From TemperatureAdjustment table
Note: If temp adjustment terms are not found in database, default values of 0.0 are used.

Output Variables:

TempAdjustment (PolProcess, Zone, Month, Hour, SourceType, FuelType)

Calculation:

TempAdjustment =
1 + TempAdjustTermA * (Temperature-75)
+ TempAdjustTermB * $(\text{Temperature-75})^2$

Step 2c: Calculate fuel adjustment

Input Variables:

FuelAdjustment (SourceType, PolProcess, FuelSubType)
MarketShare (County, Year, MonthGroup, FuelSubType)

Output Variable:

FuelAdjustmentbyType (County, Year, MonthGroup,SourceType, PolProcess, FuelType)

Calculation:

FuelAdjustmentbyType =

$$\sum_{n=1}^{no.ofFuelSubTypeswithinFuelType} \text{MarketShare} * \text{FuelAdjustment}$$

ECCP-3: Calculate running energy consumption

Step 3a: Aggregate Base Emission Rates to SourceType/ Fuel Type/Model Year/ Operating Mode level

Input Variables:

MeanBaseRate (SourceBin, PolProcess, OpMode)
From EmissionRate table
SourceBinActivityFraction (SourceType, ModelYear, SourceBin, PolProcess)
From SourceBinDistribution Table

Calculation:

MeanBaseRatebyType (SourceType, FuelType, ModelYear, PolProcess, OpMode) =

$$\sum_{SourceBin=1}^{No.SourceBinswithinFuelType,ModelYear} SourceBinActivityFraction * MeanBaseRate$$

Step 3b: Aggregate emission rates to SourceType level, Apply A/C Adjustment

Input Variables:

MeanBaseRatebyType (SourceType, FuelType, ModelYear, PolProcess, OpMode) From previous calculation
OpModeFraction (SourceType, Link, HourDay, PolProcess, OpMode) From OPModeDistribution table
ACAdjustment (Zone, Month, Hour, SourceType, ModelYear PolProcess, OpMode) From ECCP- 2a

Calculation:

Total Energy:

SourceTypeEnergy (Zone, Month, HourDay, SourceType, FuelType, ModelYear, Link, PolProcess) =

$$\sum_{OpModeBin=1}^{No.OpModeBins} OpModeFraction * MeanBaseRatebyType * ACAdjustment$$

Step 3c: Calculate Total Energy

Input Variables:

SourceTypeEnergy (Zone, Month, HourDay, SourceType, FuelType, ModelYear, Link, PolProcess) from previous calculation
SCCVtypeFraction(SourceType, ModelYear, FuelType, SCCVtype)

From SCCVtypeDistribution table
SHO (SourceType, Age, Link, Hour, Day, Month, Year)
From SHO Table
IMOBDAdjustment (SourceType, County, Year, Age, FuelType)
From IMOBDAdjustment table.
Note: If IMOBD adjustment value not found in database, a default value of 1.0 is used.
TempAdjustment (ECCP-2b, existing code)
FuelAdjustment (ECCP-2c, existing code)
PetroleumFraction (ECCP-1a, existing code)
FossilFraction (ECCP-1a, existing code)

Calculation:

Total Energy

EmissionQuant = SCCVtypeFraction *

$$\sum_{SourceBin=1}^{No.SourceTypes} SHO * SourceTypeEnergy * FuelAdjustment * TempAdjustment * IMOBDAdjustment$$

Petroleum Energy:

EmissionQuant =
EmissionQuant (Running, Total Energy)
* PetroleumFraction

Fossil Energy:

EmissionQuant =
EmissionQuant (Running, Total Energy)
* FossilFraction

ECCP-4 & 5: Calculate start and extended idle energy consumption

The equations are exactly the same as is done for ECCP-3 above. The only difference is that, under Step c, "SHO" is replaced by "Starts" for Start and "ExtendedIdleHours" for Extended Idle.

ECCP-6: Calculate total, petroleum, and fossil energy consumption for well-to-pump (WTP)

For all pollutants, well-to-pump energy and emissions are calculated as a function of pump-to-wheel total energy consumption (i.e. the sum of running, start and extended idle energy)

Step-6a: Calculate total pump-to-wheel(PTW) energy consumption

Inputs:

EmissionQuant(…FuelType, PolProcess = TotalEnergy; Running, Start, Extended Idle)

Outputs:

PTWEmissionQuant (FuelType, Total Energy)

Calculation:

PTWEmissionQuant =

EmissionQuant (Running, Total Energy)
+ EmissionQuant (Start, Total Energy)
+ EmissionQuant (Extended Idle, Total Energy)

Step-6b: Interpolate well-to-pump factorby FuelSubType in analysis year

Inputs:

WTPFactor (Pollutant, FuelSubType for next highest and next lowest in WTP factor database, relative to YearID)
(Note : WTPFactor represents the field EmissionRate located in the GreetWellToPump table of the MOVES database.)

Outputs:

WTPFactor (Pollutant, FuelSubType, YearID)

Calculation:

Linear interpolation using "bracketing" years in WTP database:
WTPFactor = WTPFactor (LoYearID) +
(WTPFactor (HiYearID) – WTPFactor (LoYearID)) *
((YearID – LoYearID) / (HiYearID – LoYearID))

Step-6c: Calculate well-to-pump factor by FuelType in analysis year

Inputs Variables:

WTPFactor (Step 6a)
MarketShare (County, Year, MonthGroup, FuelSubType,)

Output Variables:

WTPFactorbyFuelType(Pollutant, FuelType, YearID)

Calculation:

WTPFactorbyFuelType =

$$\sum_{n=1}^{no.ofFuelSubTypeswithinFuelType} \text{MarketShare} * \text{WTPFactor}$$

Step-6d: Calculate well-to-pump energy consumption

Inputs:

PTWEmissionQuant (Step 6a)
WTPFactorbyFuelType (Step 6c)

Outputs:

EmissionQuant (PolProcess = Well-To-Pump; Total Energy, Petroleum Energy, Fossil Energy)

Calculation:

EmissionQuant =
PTWEmissionQuant * WTPFactorbyFuelType

10.11. Distance Calculator

The TotalActivityGenerator produces distance information in the SHO table. The DistanceCalculator reports vehicle travel distance information based on these SHO table values, if requested by the RunSpec, in the MOVESActivityOutput table of the MOVES output database.

10.11.1. Algorithm Overview

The “total” distance in the SHO table must be disaggregated to the often more detailed level of the MOVES output. This is a “calculator-like” function and so is performed by an EmissionCalculator class despite the fact that “distance” is not a “pollutant”.

This calculator is instantiated whenever distance output is requested by the RunSpec, which implies that some pollutant-process involving the running process has been selected, and signs up for the “Running Exhaust” emission process at the “Year” level of the MasterLoop, the same level at which the TAG operates.

10.11.2. Distance calculation

If the “Distance Traveled” output is requested by the RunSpec, (which also implies that some pollutant has been selected for the Running process), the distance field in the SHO table is calculated by the TotalActivityGenerator (TAG). Otherwise distance is output by that generator as NULL.

Within the distance calculator itself, there is a single calculation step:

DC-1 : Allocate Distance to Finest MOVES Output Level

Input Variables:

SCCVtypeID(SCC) from SCC table
SCCVtypeFraction(sourceTypeModelYearID, fuelTypeID, SCCVtypeID)
From SCCVtypeDistribution table
sourceBinActivityFraction (sourceTypeModelYearID, polProcessID, sourceBinID) from SourceBinDistribution table
distance(sourceTypeID, linkID, ageID, yearID, monthID, hourDayID) from SHO table

Output Variable:

MOVESActivityOutput.distance(runID, yearID, monthID, dayID, hourID, stateID, countyID, zoneID, linkID, sourceTypeID, fuelTypeID, modelYearID, roadTypeID, SCC)

Conceptual Calculation:

MOVESActivityOutput.distance =
SCCVtypeFraction* distance *
[sum over all sourceBins in fuelTypeID (sourceBinActivityFraction)]

Several detailed considerations must be kept in mind as these calculations are performed.

One consideration is just that model years are converted to ageID values by the relationship: modelYearID = yearID - ageID.

Another consideration is that SHO table values are reduced by the SCCVtypeFraction, because the raw MOVESOutput data is for SCC-SourceUseType intersections. This factor is looked up in the SCCVtypeDistributionTable based on sourceTypeID, modelYearID, and fuelTypeID. In general this yields multiple values of SCCVtype and SCCVtypeFraction. These values of SCCVtype are used in combination with RoadType to determine SCC values from the SCC table.

The most significant complexity is that SHO table distance values are reduced by a fraction representing the portion of this "total" activity occurring for the specific fuelTypeID. This is accomplished by determining the portion of the SourceBinDistribution which involves the given fuelTypeID.

The table 10-8 illustrates the structural differences between the input to this calculation and the output it must produce:

Table 10-8. Structural Differences between DistanceCalculator Input/Output

Key Field	**SHO.distance**	**Most detailed <u>MOVESOutput</u>**
MOVESOutputRowID		Yes, autoincremented
runID		Yes, but constant for a run
yearID	Yes	Yes
monthID	Yes	Yes
dayID	Yes	Yes
hourID	Yes	Yes
stateID		Yes, but redundant with county
countyID		Yes, but redundant with zone
zoneID	Yes	Yes
roadTypeID	Yes	Yes
pollutantID		Yes, but constant for this calculation
Processed		Yes, but constant for this calculation
sourceTypeID	Yes	Yes
fuelTypeID	No	Yes
modelYearID	as ageID	Yes
SCC	No	Yes

A final consideration is that, since non-key attributes in addition to "distance" may be added to the MOVESActivityOutput table in the future (e.g. number of starts) which may be produced by other calculators, the Distance Calculator creates MOVESActivityOutput records to report its results if they don't already exist, but just updates the distance attribute in the appropriate records if they already exist.

10.12. Methane (CH4) and Nitrous Oxide (N2O) Calculator

MOVES includes two EmissionCalculator classes, CH4N2OrunningStartCalculator and CH4N2OWTPCalculator, which calculate Methane (CH_4) and Nitrous Oxide (N_2O) emissions for three processes (running, start, and well-to-pump) for each source type on each roadway type modeled in MOVES2004. These calculators use input from CMITs produced by the total activity and source bin distribution generators, and calculate the quantities of these emissions in the form required by the MOVESOutput database.

These calculators subscribe to the MasterLoop at the year level, which means they are executed once for each location (linked) for each year.

When these EmissionCalculators encounter a missing input value in the MOVES database, the result of the calculation are considered as missing. Such results are left out of the database and are not represented in the MOVES output by a value of zero or some other numeric value or character.

The calculation steps are described below. These steps are the same for CH_4 and N_2O.

10.12.1. Step 1: Calculate Running Emissions

Input:

SHO (SourceType, Age, Link, Hour, Day, Month, Year)
SourceBinActivityFraction (SourceType, ModelYear, PolProcess , SourceBin)
MeanBaseRate (PolProcess, SourceBin, OpMode)

Output:

EmissionQuant (SourceType, Pollutant, Process, FuelType….see MOVES_2004 output database design)

Calculation:

EmissionQuant = SHO * SourceBinActivityFraction * MeanBaseRate

10.12.2.Step 2: Calculate Start Emissions

Input:

Starts (SourceType, Age, Zone, Hour, Day, Month, Year)
SourceBinActivityFraction (SourceType, ModelYear, PolProcess , SourceBin)
MeanBaseRate (PolProcess, SourceBin, OpMode)

Output:

EmissionQuant (SourceType, Pollutant, Process, FuelType….see MOVES_2004 output database design)

Calculation:

EmissionQuant = Starts * SourceBinActivityFraction * MeanBaseRate

10.12.3. Step 3: Calculate Well-To-Pump Emissions

Step 3a: Calculate well-to-pump factor by FuelType in analysis year

Inputs Variables:

EmissionRate from GREETWellToPump (Year, Pollutant, FuelSubType)
MarketShare (County, Year, MonthGroup, FuelSubType,)

Output Variables:

WTPFactorbyFuelType(Year, Pollutant, FuelType)

Calculation:

WTPFactorbyFuelType =

$$\sum_{n=1}^{no.ofFuelSubTypeswithinFuelType} \text{MarketShare} * \text{WTPFactor}$$

Step 3b: Calculate Well-To-Pump Emissions

Input:

PTWEmissionQuant for total energy (Step 6a of Total Energy Calculator)
WTPFactorbyFuelType for CH4/N2O (Step 3a above)

Output:

EmissionQuant (SourceType, Pollutant, Process, FuelType….see MOVES_2004 output database design)

Calculation:

EmissionQuant = PTWEmissionQuant * WTPFactorbyFuelType

10.13. [This section number is reserved for future use.]

10.14. Result Data Aggregation and Engineering Units Conversion

This function aggregates the results produced by MOVES EmissionCalculators to the level of detail called for in the run specification and converts these results to the engineering units it specifies. Aggregation is performed to the extent possible by the MOVES Worker program and completed by the MOVES Master program. Conversion to engineering units is the final operation performed and is done by the MOVES Master program.

10.14.1. How Aggregation Levels are Specified

The level of aggregation is specified on the MOVES GUI Output EmissionsDetail Screen. These specifications are made in terms of what distinctions are desired in the output. Output rows are always distinguished by time periods. The level of this distinction may be hour, day, month or year. (Choices may be more limited, however, if time period “preaggregation” of the database was performed.) Output rows are always distinguished by location. The level of this distinction may be Nation, State, County, Roadtype or Link. (Choices may be more limited, however, if geographic "preaggregation" of the database was performed.) Output is always distinguished by pollutant.

When reporting for entire months or years, MOVES2004 scales the results up by the number of weeks in each month, i.e. by the number of days in the month divided by seven. The reader may wish to refer to section 9.2 of this document where MOVES time periods are discussed.

Output may be distinguished by SourceUseType (the recommended option), Source Code Category (SCC) or neither. (But not both since the two highway vehicle classification schemes are exclusive.)

Output may optionally be distinguished by:

Model Year
Fuel Type
Emission Process
Roadtype

In general, any combination of these distinctions may be specified. Output by SCC implies that roadtype and fueltype will be distinguished and the MOVES GUI enforces this. If "Roadtype" is selected as the Location level then "Roadtype" is automatically distinguished in the output (but the reverse is not necessarily true).

10.14.2.Aggregation Algorithm - Logical Level Specification

The raw output of MOVES is distinguished by year, month, day, hour, state, county, zone, link, road type, pollutant, process, source use type, fuel type, model year, and SCC. Taken together these fields may be considered an alternate key for the MOVESOutput table, except that, once aggregations are performed, they may assume the null value.

At macroscale, zones are redundant with counties, so whenever a countyID is present, its corresponding zoneID is also present, and when counties are aggregated out, then so are zones. Also at macroscale, links represent a combination of a county and a roadtype, so when county and roadtype are both known, so is link. Otherwise linkID is null whenever either county or roadtype is null.

The following aggregations may be carried out as implied by the RunSpec. Unless stated otherwise, any combination of these aggregations may be called for. This description is at a logical level; physical implementation differs. For example aggregation to the state level is not actually performed by aggregating roadtypes to counties and then aggregating counties to states.

a. Output is always aggregated by either SCC or source use type (or both), depending on which of these detail levels is not selected in the GUI Output Selection Panel. (The raw output is in terms of SCC-sourceusetype combinations.) The GUI enforces the constraint that at least one of these two levels of detail is not required.

b. Hours are combined into to days (of any week) if the output time period is day, month or year. A warning message is generated if this aggregation is performed and all hours are not selected in the RunSpec.

c. Days are converted to months if the output time period is month or year. The number of days in each month is obtained from the MonthOfAnyYear table, adding 1 day to February for leap years. 1/7 of these days are considered to be Mondays, 1/7 Tuesdays, etc. without regard to calendar year. It is the user's responsibility to insure that all desired days of the week are included in the RunSpec. A warning is generated if this aggregation is performed and all days are not selected in the RunSpec.

d. Months are totaled to years if the output time period is year. It is the user's responsibility to insure that all desired months are included in the RunSpec. A warning is generated if this aggregation is performed and all months are not selected.

e. Road types are combined into county totals whenever "roadtype" has not been selected in the Output Emissions detail screen. Note that selection of the SCC level of detail forces road type to be selected and this aggregation not to be performed, as does selection of the road type level of geographic output detail. A warning message is generated if this aggregation is performed and all road types are not included in the runspec.

f. State totals are combined into a single national (or total user modeling domain) result if the national level of geographic detail is selected. This aggregation is performed even if the road type level of detail has been selected on the "Output Emission Detail" panel. It is the user's responsibility to insure that all desired states are included in domain totals.

g. Fuel types are combined within source use types if this level of detail is not selected in the Output Emission Detail panel. It is the user's responsibility to insure that all desired fuel types are included.

h. Emission processes are combined if this level of detail is not selected in the Output Emission Detail panel. It is the user's responsibility to insure that all desired processes are included.

10.14.3. Engineering Unit Conversion

Engineering units are contained in the run specification for mass, energy, time and distance. In the MOVES GUI these are specified on the "Outputs" panel within the "General Output Screen". Supported are:

time units (seconds, hours, days, weeks, months and years).
mass units (kilograms, grams, pounds, and U.S. tons.)
energy units (Joules, kiloJoules, and million BTU)
distance units (miles and kilometers)

The engineering units used are reported in the MOVESRun table.

Distance units are only required if the run specification calls for travel distance to be reported.

MOVES always reports mass pollutant emission results in terms of mass per time unit and always reports energy consumptions results in terms of energy per time unit. If the time unit equals the output time period specified on the "Output Emissions Detail" screen, then the quantities reported amount to an inventory for that time period.

10.15. Post-Processing Script Execution

The "Post Processing" menu in the MOVES2004 GUI includes a selection to "Run MySQL Script on Output Database". When this item is selected MOVES searches the "OutputProcessingScripts"subdirectory in its "...database" directory, and presents a list of file names found there which have a file name extension of ".sql".

When one of these files is selected, MOVES executes the file as a MySQL script on the MOVES output database specified by the currently active run specification (or gives an appropriate error message if no output database is specified). These scripts should operate only on the currently active MySQL database, should not include the MySQL USE command, should use as input only the four tables contained in the MOVES Output database schema, (along possibly with MOVESDefault) and should store any files and tables they produce in this same database. The MOVES program, however, does not enforce these conventions.

10.16. GREET Model Interface

MOVES is designed to interface with GREET, as detailed in the following section. Further detail on the GREET interface itself is contained in a separate document entitled "User Manual and Technical Issues of GREET for MOVES Integration", prepared by Argonne National Laboratory.

10.16.1. MOVES2004 GREET Interface Functionality:

1. If requested in the RunSpec, MOVES2004 calculates well-to-pump inventories for total energy consumption, petroleum-based energy consumption, fossil fuel-based energy consumption, N2O, and CH4, using the emission factors in the GREETWellToPump table.

2. MOVES2004 offers a "preprocessing" menu option to "Update Well-To-Pump" factors. When this menu option is selected MOVES2004 produces, in XML format, a table of information needed by GREET. This contains a list of calendar years, and a list of MOVES fuel sub-types implied by the RunSpec.

3. MOVES2004 then executes the GREET GUI described below.

4. When control is returned, MOVES2004 receives a tab-separated variable, well-to-pump emission factor result table from the GREET GUI and incorporates these factors into the GREETWellToPump table in a database suitable for use as in input database by subsequent MOVES2004 model runs.

5. MOVES2004 interacts with the GREET GUI, which in turn invokes the GREET spreadsheet model. MOVES2004 does not interact directly with the GREET spreadsheet model, only with the GREET GUI.

10.16.2. GREET and GREET GUI Functionality

The design of the interface between MOVES2004 and GREET is based upon the following functional characteristics of GREET.

1. GREET estimates Well-to-Pump and Vehicle Manufacture/Disposal emissions of Total Energy Consumption, Petroleum-based Energy Consumption, Fossil Fuel-based Energy Consumption, N2O, and CH4 appropriate to calendar years 1990 thru 2050. (Internally to GREET, estimates are actually produced by interpolating between estimates applicable to five year periods and estimates beyond 2020 are based on many of the same parameter values as those for 2020.)
2. When the two models are used together, the United States (or the user modeling domain) is modeled as a single geographic region.
3. Where GREET has multiple Pathway Options for a fuel, it is generally possible for the user to supply input parameters for each pathway (with associated percentages which sum to unity), and these Pathway Options and percentages may vary by calendar year.
4. Only GREET parameters accessible from a version of the GREET GUI may be altered. To alter more deeply embedded GREET parameters the user would have to manually modify the underlying GREET spreadsheet before running MOVES.
5. GREET estimates well-to-pump emissions relative to consumption of the following fuels (in MOVES2004 these are referred to as fuel subtypes):

   Conventional Gasoline

   Reformulated Gasoline

   Gasohol (E10)

   Conventional Highway Diesel Fuel

   (GREET has the logic built in to use its high sulfur version of this fuel prior to 2006, its low sulfur version after 2006 and a blend for 2006).

   Biodiesel

FT Diesel

CNG

LPG

Ethanol

Methanol

Gaseous H2

Liquid H2

Electricity

6. The GREET GUI, while not itself written in Java, can be run from a Java program, i.e. MOVES2004. It accepts command line parameters which specify:

   - an XML input file name
   - an output file name (for well-to-pump factors and vehicle manufacture/disposal factors)
   - an error file name (used to return any GREET error messages to MOVES).

7. Execution of the GREET GUI normally results in execution of the version of the GREET spreadsheet, but may return to MOVES without executing it if the user desires. Upon return to MOVES2004, status information is available as to whether the GREET spreadsheet was run or not and whether execution was successful or an error occurred. Error messages are returned to MOVES in a separate file.

8. The GREET GUI accepts an XML-formatted list of MOVES2004 Calendar YearIDs, determines what five-year GREET periods are needed to estimate emissions for all years listed, and executes the GREET spreadsheet for each such period, and interpolates between these results as needed to produce results for the calendar years listed. GREET also accepts an XML-formatted list of fuel types (MOVES fuelSubtypes). GREET results are limited to these fuels.

9. The GREET GUI consolidates the results of these GREET spreadsheet runs into a single tab-delimited table of well-to-pump emission factors and (eventually) a single tab-delimited table of vehicle manufacture/disposal emission factors for use by MOVES2004. The table of well-to-pump emission factors is described in the next section.

10.16.3. GREET Well-to-Pump Emission Factor Result Table:

This table is a tab-separated variable ASCII text file and contains the following fields:

a. pollutantID (an Integer from the set of values used in MOVES2004)

b. fuelSubtypeID (an Integer from the set of values used in MOVES2004)

c. yearID (an Integer identifying the calendar year to which the factor applies)

d. emission rate (a floating point number, expressed in the appropriate units)

Fields a, b, and c, together form the primary key of this table. Field d is its only non-key field. The fields pollutantID, fuelSubtypeID and yearID are as defined in the MOVESDefault database schema.

10.17 Future Emission Rate Creator (FERC)

The Future Emission Rate Creator (FERC) allows MOVES users to alter the energy and emission rates for advanced technology vehicles for model years 2001 and later, and for all energy and emission rates (conventional and advanced technology) for 2011 and later. The FERC works with user-supplied ratios which express the relative benefit of advanced technologies versus conventional technology, by operating mode, to generate advanced technology rates. The FERC is an "external control strategy", meaning it requires work outside of MOVES with MySQL databases and is executed from the "Pre-Processing" menu of the MOVES GUI, rather than being executed as part of the model run itself.

Input data: The FERC requires two input tables, one to create "short term" future rates and another to create "long term" future rates. These are stored in ASCII text, comma separated variable (CSV) format. To generate an alternative set of future emission rates the user must alter these tables directly (outside of MOVES) before running the FERC. The user names these tables as desired and selects them from the MOVES GUI. At least one set of example input tables will be supplied with MOVES.

The tables contain identical fields, except that the opModeID field is present only in the "short term" table. The first record in each file contains column names to facilitate human readability and is ignored by the calculations. The FERC input table fields are shown in Table 10-9.

Table 10-9: FERC Input Table Fields

Field	Description
polProcessID	Pollutant / Process ID 501 = CH4 running 502 = CH4 start 601 = N2O running 602 = N2O start 9101 = total energy running 9102 = total energy start 9190 = total energy start
opModeID (present in short term table only)	Operating Mode ID 0-36 = running VSP/speed modes (total energy only) 200 = start mode

	300 = running mode (CH4 and N2O)
targetFuelTypeID	Type of fuel: 1 = Gasoline 2 = Diesel 3 = CNG 4 = LPG 5 = E85 6 = M85 7 = Gaseous Hydrogen 8 = Liquid Hydrogen 9 = Electricity
targetEngTechID	Engine Technology ID 1 = Conventional Internal Combustion (CIC) 2 = Advanced Internal Combustion (AIC) 11 = Moderate Hybrid CIC 12 = Full Hybrid CIC 20 = Hybrid AIC (used only for Hydrogen IC hybrid) 21 = Moderate Hybrid AIC 22 = Full Hybrid AIC 30 = Electric 40 = Fuel Cell 50 = Hybrid Fuel Cell
targetModelYearGroupID	Model Year Group ID 20012010 = 2001 thru 2010 20112020 = 2011 thru 2020 20212050 = 2021 thru 2050
fuelTypeID	Base fuel type ID (same as targetFuelTypeID)
engTechID	Base engine technology ID (same as targetEngTech ID)
modelYearGroupID	Base model year group ID (same as targetModelYearGroupID)
fuelEngAdjust	Fuel / Engine Adjustment, i.e. the relative benefit of the target fuel / engine technology versus the base fuel/engine technology. A value of 1 = no benefit, a value of 0.5 = 50% improvement
dataSourceID	5001 thru 5004 = FERC short term 6000 = FERC long term

These fields fall into three functional groups:

1. fields describing the emission rates produced by the adjustment
    a. polProcessID
    b. opModeID
    c. targetFuelTypeID
    d. targetEngTechID
    e. targetModelYearGroupID
    f. dataSourceID

2. fields describing the base technology emission rates to which the adjustment is applied
    a. polProcessID
    b. opModeID
    c. fuelTypeID
    d. engTechID
    e. modelYearGroupID

3. the adjustment itself
    a. fuelEngAdjust

Of these fields, the user would generally only need to alter values of fuelEngAdjust, which are the ratios which define the benefit of the target technology relative to the base technology. Note that the values of polProcessID and opModeID contained in the base records are carried into the future emission rates produced, as are the values of regClassID, engSizeID and weightClassID implicit in the base record sourceBinIDs.

The short term table contains the information necessary to produce alternative fuel and advanced vehicle technology rates for model years 2001 through 2010 (which are treated as one model year group). The long term table contains the information necessary to produce conventional and advanced vehicle technology rates for model year groups 2011 and later, split into two model year groups: 2011 - 2020 and 2021 - 2050. The main difference between the tables is that the short term table uses as base rates the gasoline and diesel rates for conventional technology in the 2001 through 2010 model year group, while the long term table uses the 2001 – 2010 rates as a base the 2001-2010 rates for each technology (i.e. those generated by the short term table). The long term table enables the user to model technology evolution over time, within each technology.

Output produced:

The FERC produces a MySQL database suitable for use as a MOVES2004 user input database. This database contains two tables, EmissionRate and SourceBin. The name of this database is specified by the user.

Calculations performed:

Short term future emission rates are created by applying fuelEngAdjust as a multiplicative adjustment factor to the meanBaseRate of all non-motorcycle EmissionRate table records in the MOVESDefault database with dataSourceIDs less than 5000. For each such record with matching values of polProcessID, opModeID, fuelTypeID, engTechID, and modelYearGroupID a short term future emission rate record is generated having the targetFuelTypeID, targetEngTechID, targetModelYearGroupID and new dataSourceID (while retaining the prior values of polProcessID, opModeID, regClassID, engSizeID, and weightClassID).

Long term future emission rates are created by applying fuelEngAdjust as a multiplicative adjustment factor to the meanBaseRate of all EmissionRate table records in the MOVESDefault database with dataSourceIDs less than 5000 (including those for Motorcycles), along with the short term future emission rates produced by the first step. For each such record with matching values of polProcessID, fuelTypeID, engTechID, and modelYearGroupID a long term future emission rate record is created having the targetFuelTypeID, targetEngTechID, targetModelYearGroupID, and new dataSourceID (while retaining the prior values of polprocessID, opmodeID, regClassID, engSizeID, and weightClassID).

For this calculation to operate correctly the MOVESDefault.SourceBin table must contain a record for each sourceBinID present in the EmissionRate table having a dataSourceID less than 5000. The MOVESDefault database distributed with MOVES2004 satisfies this condition.

10.18 EPA's Plans to Estimate the Uncertainty of MOVES Results

MOVES2004 does not include the ability to estimate uncertainty in model estimates. The estimation of uncertainty was strongly considered for inclusion, and significant effort was made to assess how either Analytic Propagation of Error or Monte Carlo methods could be used to estimate uncertainty; but we decided to defer the inclusion of uncertainty to allow more time to resolve issues with computation methods, model performance and to better estimate the uncertainty of model inputs. This section details our thinking on uncertainty up to this point, and design concepts for inclusion in future versions.

Uncertainty is introduced by the model theory, the mathematical formulation of the model, and the data used to calibrate the model. The first two sources of uncertainty are not amenable to quantification, so the uncertainty estimates in MOVES will focus on the data used to calibrate the model. However, even with this narrower focus, the challenges are significant. MOVES uses data for hundreds of variables covering all aspects of mobile source emission generation: the vehicle fleet, vehicle activity patterns, emission rates, fuel properties, geographic location, fuel properties, and meteorology. The uncertainty of emission rates is readily quantified using standard data analysis techniques, but much of the fleet, activity, fuel and meteorology data sources have limited information on the full range of uncertainties. The primary challenge in including uncertainty in MOVES is therefore to quantify uncertainties for all of the input data used in the model.

We plan to produce uncertainty estimates using Monte Carlo (MC) methods. As an alternative, we have examined analytical propagation of error (APOE), which has the advantage of providing an answer following a single model run. It has the disadvantages of being an approximate method, of being programming-intensive, and of providing no information regarding the form of the output distribution. MC has the advantages of accuracy (if iterated enough), simplicity, extendibility, and detailed information on output distribution, but has the disadvantage of requiring large numbers of model iterations. Our initial explorations with APOE have revealed problems related to modular software design and distributed processing. Monte Carlo methods avoid these problems but require that the generation of Monte Carlo inputs be done in a centralized fashion. Since Monte Carlo analyses of large simulations are likely to be computationally prohibitive, we are exploring the possibility of estimating the uncertainties of large simulations by running smaller simulations.

To fully implement MC simulation, for each uncertain data point we would add to the database the distribution of the mean, including the type of distribution and its defining parameters. For a non-Monte Carlo model run (i.e. if the user just desired a point estimate without uncertainty estimation), the mean values would populate the Execution Database, just as they do now. For a Monte Carlo simulation, the model would be run a specified number of times. For each run, the Execution Database would be populated with random samples from the distributions of the means of the data points. All output values could be saved for subsequent analysis (storage-intensive option) or summary statistics could be accumulated as the runs progress (non-storage-intensive). Confidence intervals and other statistics would be generated from these model iterations. Dynamic sensitivity can be calculated from Monte Carlo simulations by regressing model inputs against model outputs. Inputs with the highest (in absolute value) regression coefficients will be those for which the greatest improvement will result from decreasing their uncertainty.

As an intial step towards the MC approach described above, the MOVESDefault database currently includes a Coefficient of Variation (CV) field for most input fields. Most of the these fields have been left intentionally blank, to be used as placeholders for when uncertainty is added; although many records in the EmissionRate table have valid CV's, since they are computed by the emission rate "binner" program described in the MOVES2004 Energy and Emission Input report. Including only CV in the input databases would facilitate the approach for MC simulation using an assumed distribution form for all input variables – i.e. either normal or lognormal. To provide more flexibility in choosing distributions, the input databases would need to be expanded to include distributional form, and if distributions such as Weibull or Gamma were desired, to add an additional coefficient term.

To produce random variates that are components of distributions that must sum to one, each term of the distribution will first be generated individually. Then the resulting set of variates will be multiplied by the same factor so that the total sums to one. This procedure will result in the terms of the distribution being negatively correlated, as they should be: if one term is especially large, others must be correspondingly small, and vice versa.

The nature of emissions calculations lends itself to distributed processing (e.g., the emissions in one county do not affect the emissions in another). The MOVES design takes advantage of this fact by allowing the distribution of independent calculations over many machines. Both modular design and distributed processing make use of the results of independent modules in subsequent calculations without having to know the details of how they were calculated. For example, when the emissions of several counties for a given time period have been produced on a number of separate machines, we can simply add the emissions, without knowing how they were calculated. This scheme creates no problem for APOE as long as there are no common random variates in all the prior independent calculations. Then, just as we can work with the results of prior calculations without knowing their details, we can also work with the variance of those prior results without knowing their details. Such a scheme allows APOE to be easily applied to any set of calculations no matter how they are broken up. For example, the variance of the sum of a set of county emissions will simply be the sum of the variances of the county emissions.

However, the requirement that there are no common random variates in all the prior independent calculations is generally not met in the MOVES model. Many of the same random variates (e.g., emission factors, deterioration factors, and temperature corrections) are likely to be used in multiple counties. Then the variance of the sum of a set of county emissions will no longer simply be the sum of the variances of the county emissions. The presence of a common random variate in the separate calculations creates a kind of computational correlation that increases the variance of the sum. In order to correctly use APOE to calculate the variance of the sum (or other function) of previously calculated emissions, the calculation itself must be performed in a centralized fashion so that the partial derivatives of the sum with respect to each common random variate can be determined for the whole calculation. (This result may be verified with a sample APOE calculation.) Every final result from the model must then be done in a single step, so that the partial derivatives of the final result with respect to each contributing parameter can be correctly determined. This problem is the basic reason that we are not using analytical propagation of error.

A similar issue exists for Monte Carlo simulations, but can be handled without defeating the modularity of the calculation. For each iteration, the realization of each random variate must be the same everywhere it is used. We plan to accomplish this by producing a new Execution Database for each Monte Carlo iteration. All parameters used in calculations are distributed to various workers or modules from the execution database. Every parameter then has a single value for a given iteration that is identical everywhere it is used.

11. MOVES2004 Input and MOVESDefault Databases

The principal input data for a MOVES model run is normally obtained from the MySQL database named MOVESDefault. A version of this database is included in each distribution of MOVES. The user may change this database if desired. This is not normally done directly, however, unless data deletions are required, or a fundamental change to the scope of the model is involved, because MOVES2004 has a more convenient mechanism for the user to supply additional or alternative input data. The MOVES GUI program and MOVES run specifications allow the user to specify one or more MySQL databases whose data is to add to or replace the data in MOVESDefault. The simplicity and generality of this scheme is that these MOVES input databases have exactly the same table structure as MOVESDefault (or any portion of it), and may be used to replace all input data items. A limitation of this scheme, however, is that construction of these input databases is not always easy and the user is responsible for the accuracy, completeness and consistency of the new database. This is not always a simple endeavor. EPA envisions that "data importer" programs will be written to help prepare MOVES input databases and support the principal specialized input use cases as they arise. It is also envisioned that organizations outside EPA will also produce data importers for MOVES. The initial release of MOVES2004, however, does not include any data importers.

Since the MOVESDefault database and MOVES input databases have the same table structure, this structure will be referred to in the remainder of this section as simply the "MOVES Database." The MOVES Database, like all MySQL databases, is a relational database, which means (among other things) that it is made up entirely of tables, and that every record within each of these tables has the same set of data items or "fields." The MOVES database has a naming convention that table names begin with a capital letter and field names begin with a small letter (unless their name starts with an acronym).

The MOVES database structure is highly "normalized" which means that data is contained in many separate tables, several of which usually need to be joined together to satisfy an information requirement.

11.1. Use of Data Types

The overall approach to the use of MySQL column types in the MOVES database is to use integer type columns (2-byte integers where possible, 4-byte integers otherwise) for all key identifying fields (stateID, countyID, hourID, sourceTypeID, etc), and to use normal-precision floating point columns for numeric information. Since MySQL has no boolean (logical true/false) column type, the MOVES database uses columns of character type with length 1 for data items that have a yes/no or true/false nature. Such columns are populated with “Y” to represent “yes” or “true,” and “N” to represent “no” or “false.”

11.2. Functional Types of Tables:

While the MOVES Database, like all relational databases, consists entirely of tables, these fulfill several different functions. Some tables function merely to establish value lists for some of the fundamental entities in the database. Examples of such tables include State, Year, and DayOfAnyWeek.

A few tables represent “Associations” between database entities. In the MOVES database these tables usually have “Assoc” as the last part of their name. An example of an association type table is PollutantProcessAssoc, which contains information as to which pollutants are emitted by which emission processes. Since this is a many to many relationship (i.e. several pollutants are generally produced by each emission process and a pollutant can be produced by several emission processes), an “Association type” table is used to store the valid combinations.

The most common kind of table in the MOVES database stores the substantive subject matter information, and in this document are termed “information” tables. The EmissionRate table for example stores emission rates for all emission processes in MOVES2004 except well-to-pump.

Some “information” tables store data “distributions,” i.e. sets of fractions which add up to unity. The MOVES database has a naming convention that such tables have the word “Fraction” or “Distribution” as the last part of their name, and the field containing such fractions has the word “Fraction” as the latter part of its name. An example of this is the “HourVMTFraction” table whose “hourVMTFraction” field stores information as to what fraction of certain VMT occurs during each of the 24 hours of the day. Another example is the “RoadTypeDistribution” table whose “roadTypeVMTFraction” field stores information as to what fraction of certain VMT occurs on each RoadType.

Another kind of “information” table which merits special consideration consists of those which are written by MOVES Generators and which MOVES EmissionCalculators, (which can be considered to implement the MOVES Core Model), use as their principal inputs. These are called “core model input tables” or CMITs. The reason CMITs are important to the MOVES user is that they are alternative points for data entry to the model. The user has a choice as to whether to supply input data to a Generator and have the Generator populate its CMIT tables during model execution, or to place input data directly into the CMIT, (in which case the Generators are programmed not to modify it). A combination of the two approaches may also be used. Most CMITs are information type tables, but there is one notable exception to this, namely SourceBin, which can be, considered a category or an association type table.

These various kinds of tables are not always completely distinct. The SourceUseType table for example can be considered a “category” type table in that it defines the set of sourceTypeIDs available for use in the database, but can also be considered in “information” table, since it contains several subject matter information fields, e.g. sourceMass.

The remainder of this section diagrams and briefly describes the role of MOVES database tables in terms of functional table groups. These functional groups correspond to model components such as a Generator or the Core Model. Within each group the information type tables are generally discussed in the order in which their input data is used. Category and Association type tables are introduced as needed to provide the data framework for information type tables. The discussion of tables is relatively brief in this document and is only intended to give a general idea of how each table is used and its most significant characteristics. Users can refer to the MOVES database documentation, contained in the …readme/Moves.rtf file within the actual database directory, for more detailed table and field level information.

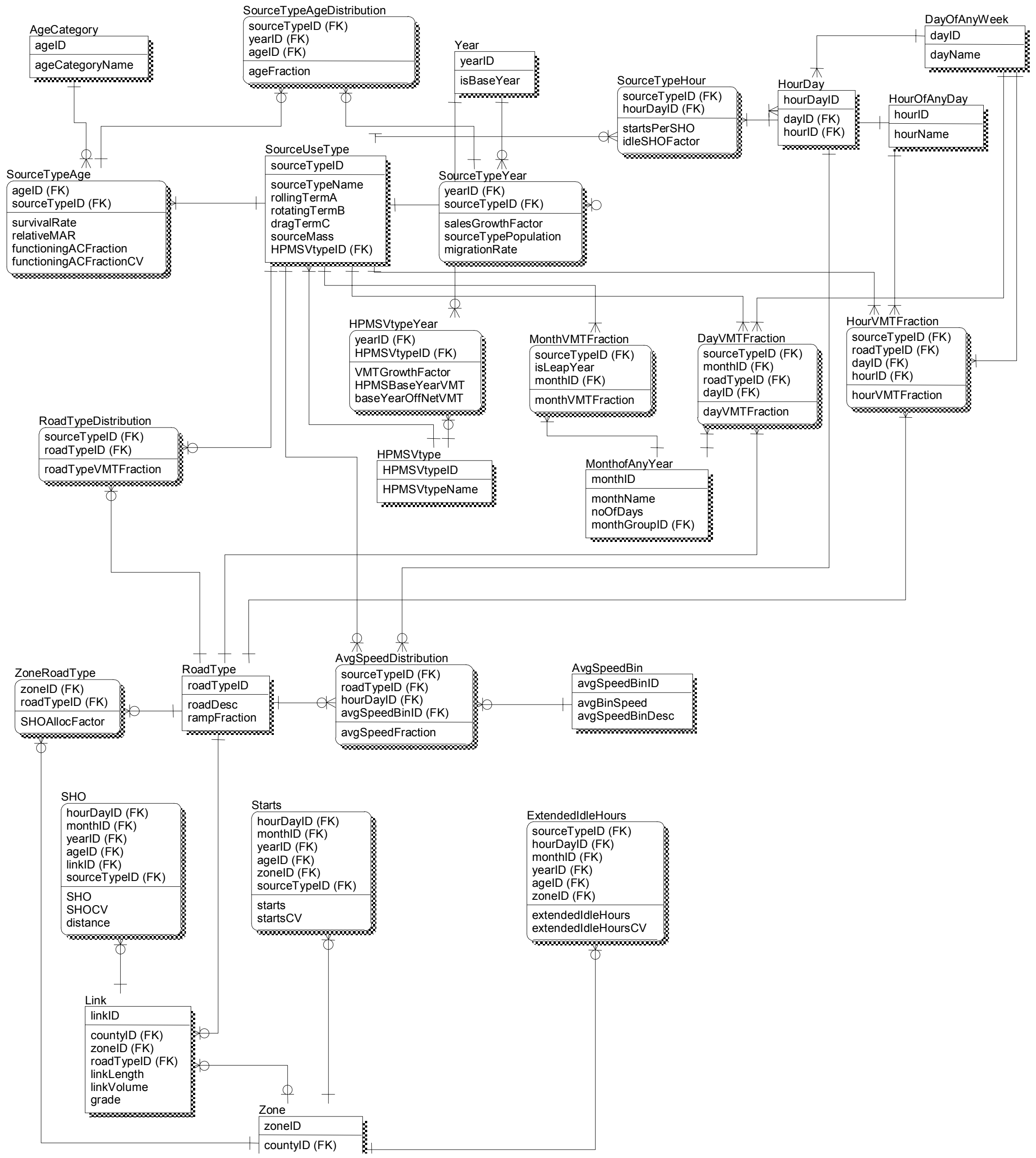
The table diagrams in the following sections of this chapter are technically termed "entity-relationship" diagrams. The graphical conventions which apply to these tables include:

- That each rectangle represents a table.
- That the primary key fields of the table are shown above the line within these rectangles
- Fields designated "FK" are foreign key fields; i.e., they help identify related records in other tables.
- Lines connecting the rectangles represent relationships between them.
- A short perpendicular line at the end of relationship indicates that it is possible that one record from that table participates in the relationship
- A small "o" at the end of a relationship line indicates that it is possible that no records from that table participate in the relationship
- A small "v" (sometimes refered to as "crow's feet") at the end of a relationship line indicates that it is possible that many records from that table participate in the relationship

11.3. Tables Related to the Total Activity Generator (TAG):

Figure 11-1 shows the tables most directly associated with the TotalActivityGenerator and the relationships between them. This is the largest table grouping, but its structure presents little difficulty.

Figure 11-1. Tables related to the Total Activity Generator (TAG).



SourceUseType is a both category table that defines the values of sourceTypeID, and a data type table that provides certain information about vehicles of each type. The MOVESDefault database defines 13 sourceTypeIDs for the "MOVES types" shown in Table 1 above. SourceUseTypes are often referred to more succinctly as

SourceTypes or UseTypes. They are considered different categories because they are expected to have distinctive usage patterns. Each SourceUseType belongs to an HPMSVtype as indicated by the HPMSVtypeID field.

Year is essentially a category table that defines the calendar years for which the model may be run. It also establishes which years are considered “base years” for the purpose of VMT and population data purposes. In MOVES2004, 1999 is the only base year in the default database.

SourceTypeYear is an information table containing population, sales growth, and migration rate information used by the TAG.

AgeCategory is a category type table that defines the valid age categories of SourceUseTypes. Each such category has an integer ageID and a character ageCategoryName. The default database defines 31 age categories where ageID=0 represents new vehicles thru age category 30 which represents vehicles which are 30 years old or older. It is unlikely the user would change this table. MOVES2004 makes assumptions about the ageID values, e.g. that modelYearID = calendar year – ageID and that ageID = 0 (new vehicles) exists.

SourceTypeAgeDistribution is a distribution information type table that stores a distribution of each SourceType into AgeCategories for each calendar Year. (This is similar to the “registration distribution” information in MOBILE6.)

SourceTypeAge is an information type table that contains data items which pertain to an AgeCategory of a SourceUseType.

HPMSVtype is a category type table that defines the Vehicle Types as classified by the Highway Performance Management System (HPMS).

HPMSVtypeYear is an information type table that contains vehicle mileage traveled (VMT) data pertaining to an HPMS vehicle type in a given calendar year, including VMT growth rates.

RoadType is essentially a category table that defines the roadTypeIDs used by the model. In MOVES2004 the default database adopts the 12 highway road types used in HPMS, along with a thirteenth value of 1 to represent off network locations. The isRamp attribute is used by the OMDG.

RoadTypeDistribution is a distribution information type table that contains the distribution of VMT across the RoadTypes.

MonthOfAnyYear is a category type table that establishes the monthIDs for the 12 months of the year. In the default data month ID = 1 is January and monthID = 12 is December.   MOVES2004 assumes that there are 12 months in the year and it is unlikely the user would change this table.

MonthVMTFraction is a distribution information type table that contains a distribution of each sourceTypeID's VMT across the MonthsOfAnyYear. Separate distributions are stored for leap years and non-leap years.

DayOfAnyWeek is a category table that establishes the dayIDs for the seven days of the week. In the default data dayID = 1 represents Mondays and dayID = 7 represents Sundays. MOVES2004 assumes there are seven days in every week, and it is unlikely the user would change this table.

DayVMTFraction is a distribution information type table that distributes the VMT for a sourceTypeID in a MonthID, on a RoadTypeID across the days of the week.

HourOfAnyDay is a category table that establishes the hourIDs for the seven days of the week. In the default data hourID = 1 begins at midnight. MOVES2004 assumes there are 24 hours in every day and it is unlikely the user would change this table.

HourVMTFraction is a distribution information type table that distributes the VMT for a sourceTypeID on a RoadTypeID on a dayID across the hours of the day.

AvgSpeedBin is essentially a category type table that defines the avgSpeedBinIDs used by the model. Its avgBinSpeed field does also function as a data item in the model.

HourDay is a category table that exists only because of database design considerations. A number of tables in the MOVES Database use both the hourID and dayID as key fields. This table defines combined hourID and dayID values, to reduce the number of key fields in these other database tables.

AvgSpeedDistribution is a distribution information type table that contains a distribution of time spent in average speed bins, by each sourceTypeID, on each roadTypeID, in each hour and day.

Zone is a category type table that establishes the zoneIDs used by the model and identifies the single county to which the Zone belongs. Although the name of this table does not include "Distribution" or "Fraction" this table is also in effect a distribution information type table that stores distributions of start and idle activity to Zones. Both StartAllocFactor and idleAllocFactor should sum to unity across all the Zones in the model domain.

ZoneRoadType is a distribution information type table that stores a distribution of source hours operating (SHO) across the RoadTypes in a Zone. (It could as easily been called SHODistribution, or, because at macroscale a combination of a RoadType with a Zone is a Link, it could have been combined with the Link table.)

SourceTypeHour is an information table that stores activity ratio information pertinent to an sourceTypeID during each HourDay.

SHO is a CMIT information type table, containing distance traveled and source hours operating (SHO) information, that may be produced by the TAG or entered directly by the user. It is worth noting in the diagram that SHO is geographically "link-based."

Starts is a CMIT information type table, containing information about the number of starts, that may be produced by the TAG or entered directly by the user. It is worth noting in the diagram that start activity is geographically "zone-based."

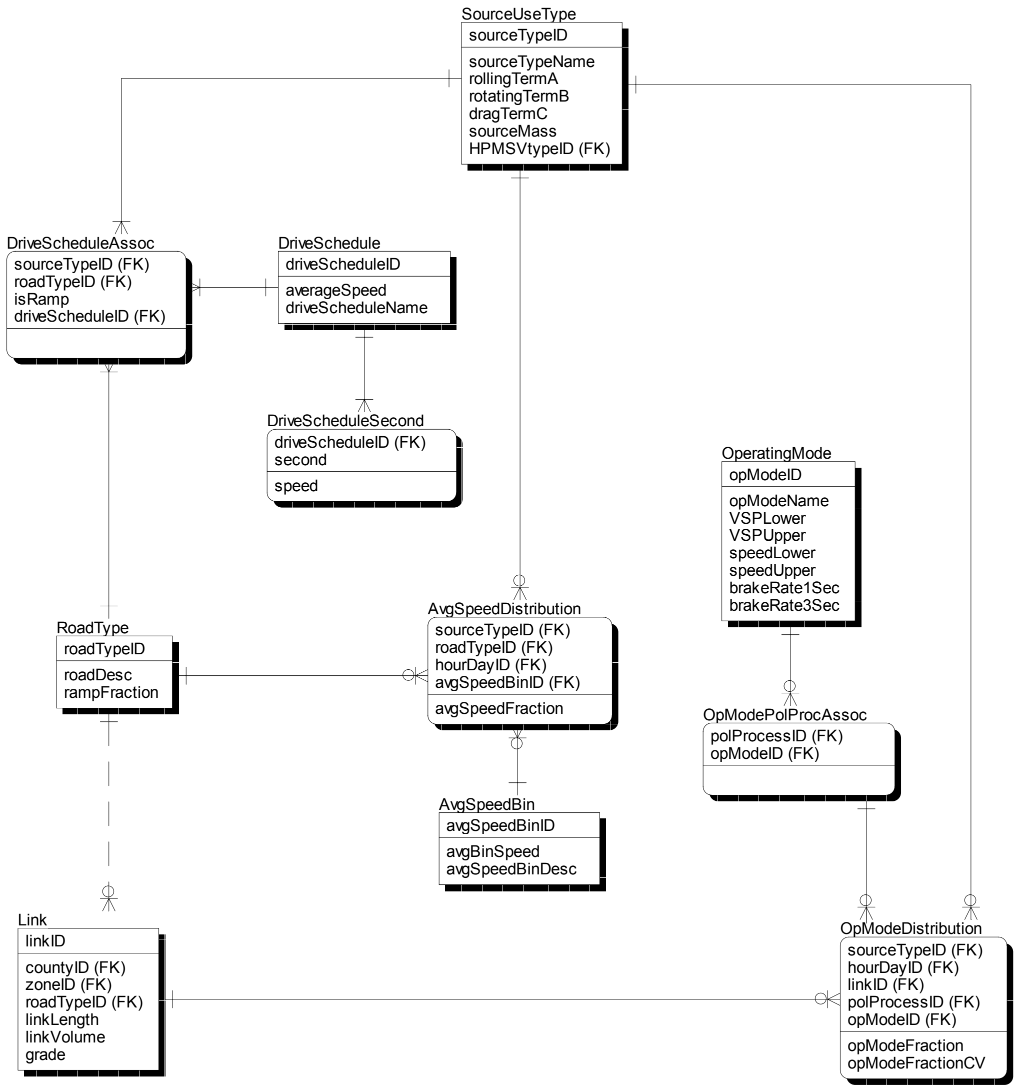
ExtendedIdleHours is a CMIT information type table, containing information about the number of extended idling hours, that may be produced by the TAG or entered directly by the user. It is worth noting in the diagram that extended idling activity is geographically "zone-based."

11.4. Tables Related to the Operating Mode Distribution Generator (OMDG):

Figure 11-2 shows the group of tables most directly related to the OMDG and the relationships between them. The structure of this group of tables is designed to solve several rather specialized problems:

how to store driving schedules in a relational database,
how driving schedules are associated with source types, road types, and average speeds in MOVES2004,
a special treatment of freeway ramp driving, and
the fact that operating mode distributions are specific to both pollutant and emission process in MOVES2004.

Figure 11-2. Tables related to Operating Mode Distribution Generator (OMDG)



AvgSpeedBin, **AvgSpeedDistribution** , **SourceUseType**, and **RoadType** have already been described.

DriveSchedule is essentially a category table which defines the set of driving schedules, segments of typical vehicle operation, used by the model. There is a single record in this table for each driving schedule which contains information about the schedule as a whole. The average speed attribute is used as a data item.

DriveScheduleSecond is an information table that contains the actual second by second speed data for each driving schedule, one record for each second. Schedules are considered to start at second = 0. There may be gaps in the second-by-second information. Such gaps divide the schedule into "snippets."

DriveScheduleAssoc is an association table that associates a driving schedule with a combination of sourceTypeID, roadTypeID, and an indication as to whether or not the schedule represents operation on freeway ramps.

Link is essentially a category table that establishes the linkID values used in the database and relates Links to Counties, Zones, and RoadTypes. Its linkLength and linkVolume fields are not used in MOVES2004. Its grade field is used by the OMDG's calculation of vehicle specific power (VSP), but the value of grade is 0.0 in the default database supplied with the model.

OperatingMode is a category table that establishes the opModeIDs used in the database. It is also an information table whose data fields are used by the OMDG.

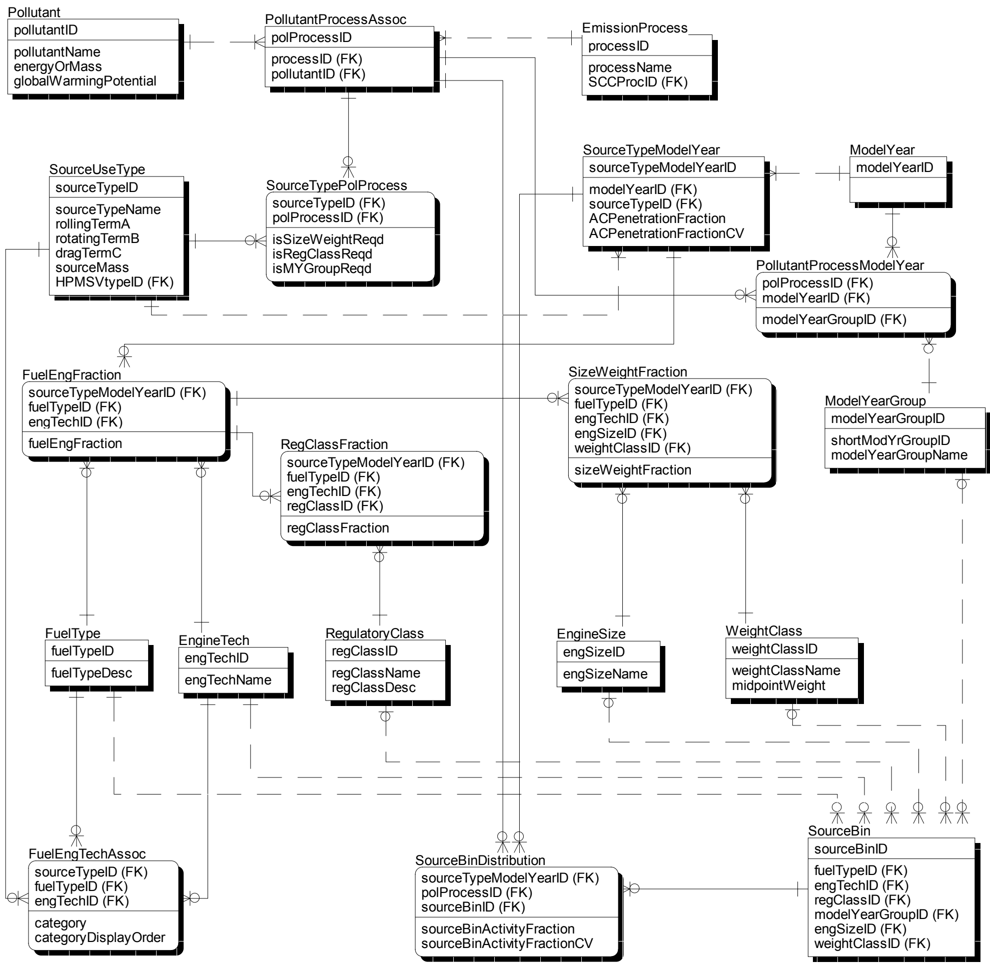
OpModePolProcAssoc is an association table that associates operating modes with pollutant-process combinations, (which are themselves an association of a pollutant and an emission process). The "starting" operating mode for example is associated only with pollutant-processes which include the "start" emissions process.

OpModeDistribution is a CMIT information table used to store the operating mode distributions calculated by the OMDG. The fact that it is a CMIT table means that operating mode distributions can also be entered directly into this table.

11.5. Tables related to the SourceBinDistributionGenerator (SBDG) and the Alternative Vehicle and Fuel Technology (AVFT) Strategy:

Figure 11-3 illustrates the group of tables most directly related to the SBDG and the relationships between them. The additional table used by the AVFT Strategy is also included in this group because of its close relationship to the SBDG. The complexity of the diagram reflects the demanding role that this portion of the database plays. This is where the source use type activity information (generally collected in terms of categories which differ from those which classify emission rates) is put into a form which can be related to the emission rate information. Additional complexity is caused by the necessity of having the SourceBin table to reduce the multiplicity of key table fields and to reduce database size. It should be noted that the MOVES2004 database schema allows some source bin discriminating factors such as RegulatoryClass, EngineSize, etc., to assume a null value.

Figure 11-3. Tables related to SBDG and AVFT Strategy.



SourceUseType has been discussed in previous sections.

Pollutant is essentially a category table that establishes the pollutantID values used in the database. It also identifies whether the pollutant is a "mass" or an "energy" type emission. The globalWarmingPotential field is not currently used.

EmissionProcess is a category table that establishes the processID values used in the database.

PollutantProcessAssoc is an association table that establishes which pollutants will be modeled for each emission process.

SourceTypePolProcess plays a fundamental role in the SBDG. First it establishes which PollutantProcesses require source bin distributions. (In this sense it functions as an association type table.) The total energy – running pollutant-process, for example, requires source bin distributions, but the total energy – well-to-pump pollutant-process does not because well-to-pump emissions are derived from the emissions of other processes.

Then for each combination of sourceBinID and PollutantProcess requiring source bin distributions, the fields of this table establish whether or not engine size and vehicle weight, regulatory class, and model year group are used to discriminate source bins. (In this role it functions as an information type table.) Engine size and vehicle weight are either both used or both not used since they are highly correlated. Fuel type and engine technology are used in all source bin distributions.

Source Bin Discriminators. Several tables serve to establish the categories used to form source bins. These are referred to in this document as "source bin discriminators." Their categories need not be mutually exclusive, though of course the set of categories used in each particular source bin distribution must be. Some discriminators are allowed to assume a value meaning "doesn't matter." These characteristics make the information in these tables useless for most purposes other than generating source bin distributions.

FuelType establishes the set of fuelTypeIDs used in the database. FuelTypes are used in several areas of the model besides the SBDG. fuelTypeID is used in all source bin distributions.

EngineTech establishes the set of engTechID values used in the database. engineTechID is used in all source bin distributions.

RegulatoryClass establishes the set of regClassID values used in the database. Whether RegulatoryClass is used in a source bin distribution is governed by the "isRegClassReqd" field of the SourceTypePolProcess table.

EngineSize establishes the set of engSizeID values used in the database. Whether engSizeID is used in a sourceBinDistribution is governed by the "isSizeWeightReqd" field of the SourceTypePolProcess table.

WeightClass establishes the set of weightClassID values used in the database. Whether weightClassID is used in a sourceBinDistribution is also governed by the "isSizeWeightReqd" field of the SourceTypePolProcess table.

ModelYearGroup establishes the set of modelYearGroupID values used in the database. Whether it is used in a sourceBinDistribution is governed by the "isMYGroupReqd" field of the SourceTypePolProcess table.

ModelYear is a category table that established the set of modelYearIDs used in the database.

SourceTypeModelYear serves in the SBDG merely to establish a single key field, sourceTypeModelYearID, that can be used in other tables in place of both sourceTypeID and modelYearID, thereby reducing the number of key fields needed in several of the other tables. Its fields related to air conditioning are used as information items by the core model.

FuelEngFraction is a distribution information table used by the SBDG to construct source bin distributions.   It contains distributions of each relevant sourceTypeModelYearID by fuelType and EngineTech.

RegClassFraction is also a distribution information table used by the SBDG. It further distributes elements of FuelEngFraction distributions by RegulatoryClass.

SizeWeightFraction is also a distribution information table used by the SBDG. It further distributes elements of FuelEngFraction distributions by engineSize and weightClass.

PollutantProcessModelYear plays an associative role between three entities: PollutantProcesses (which themselves are an association), ModelYears, and ModelYearGroups. Intuitively one would expect ModelYears to simply be grouped into ModelYearGroups. The situation is more complex however as reflected by this table. ModelYearGroup membership is allowed to vary by PollutantProcess. The "shortModYearGroupID" field of this table plays an internal role in the creation of sourceBinID values and would not normally be an end user concern.

SourceBin is a CMIT table which can be produced in whole or in part by the SBDG. As mentioned previously, it is the only CMIT in the MOVES database that is not an information type table.   The fact that sourceBin categories can be created dynamically during model execution is probably the most complicated aspect of the MOVES data structure. SourceBin contains a list of sourceBinIDs used in sourceBinDistributions. It can also be considered to establish an association between the six Source Bin Discriminators, establishing in effect which combinations are used in the model run.   Like other CMITs it can be populated directly or by a generator, in this case by the SBDG. In the MOVESDefault database distributed with the model, it contains the set of sourceBinIDs that result from running the SBDG on the default input data.

SourceBinDistribution is a CMIT information table used to store the sourceBin distributions calculated by the SBDG.   The fact that it is a CMIT table means that sourcebin distributions can also be entered directly into this table. The distributed version of the MOVESDefault database relies on the SBDG to generate all sourceBin distributions.

FuelEngTechAssoc is essentially an association type table used by the AVFT Strategy. It establishes the combinations of fuelTypeID and engTechID that are valid for each sourceTypeID.   Its information fields are used by the AVFT to help display these combinations in the MOVES GUI.

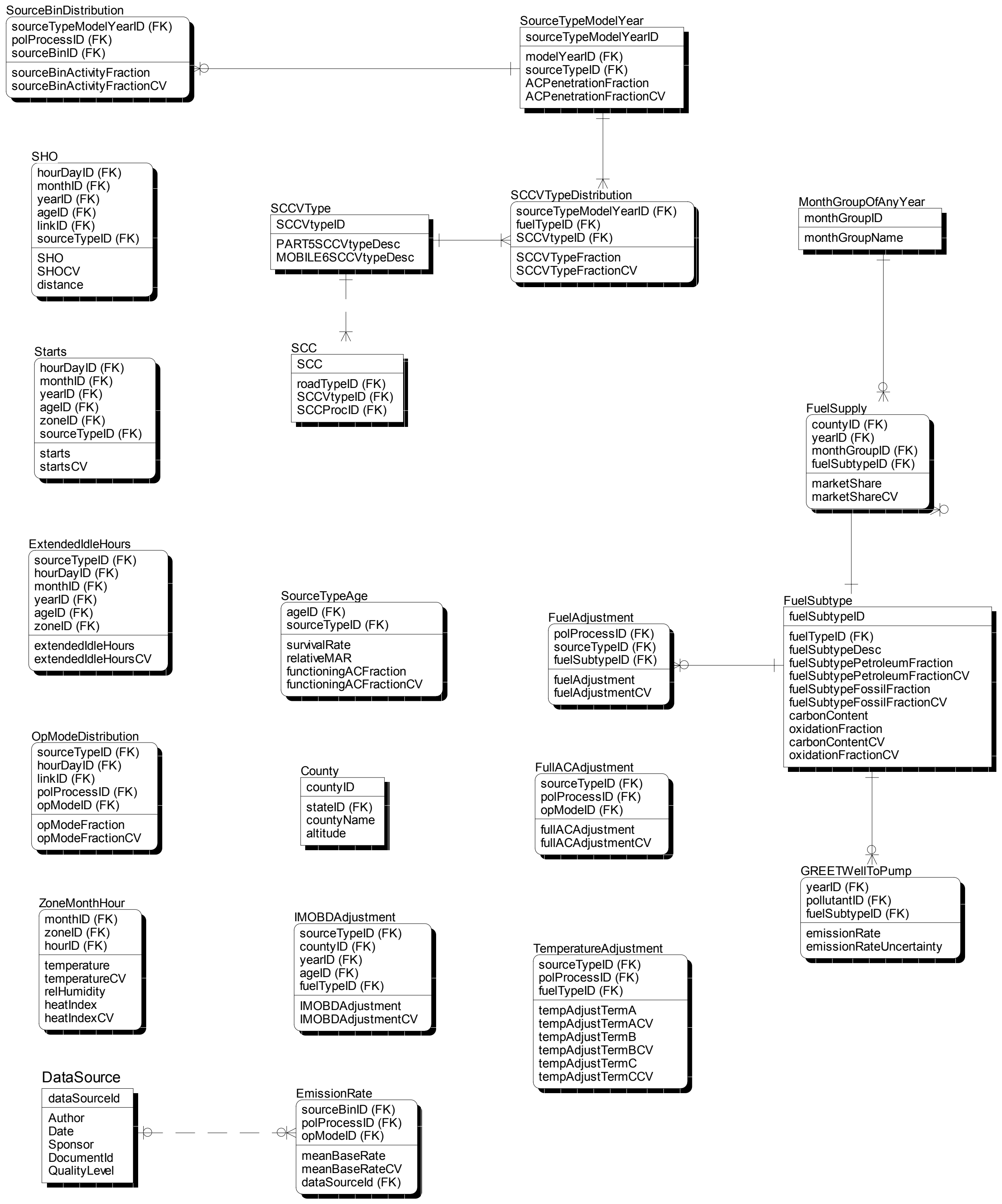
11.6. Tables Related to the MOVES Core Model

The MOVES Core Model consists of several MOVES "Calculators" which produce emission results for portions of the model run using the MOVESExecution database once "Strategies" and "Generators" have executed for these portions of the model run. Figure 11-4 shows the MOVES Database tables used by the MOVES Core Model. While the core model input tables produced by the Generators provide their principal input, the EmissionCalculators also utilize a variety of additional information from throughout the database. So the group of tables related to the Core Model does not all fit into a structure.

Several of the tables in this group pertain to vehicle fuels and the supply of fuel available in a geographic location (County). The original MOVES design envisioned that a Fuel Supply Generator program would be part of MOVES, but this has proven unnecessary in MOVES2004.

Several other tables in this group are used to produce output in terms of 144 Source Classification Codes (SCCs) which pertain to highway vehicles.

Figure 11-4. Tables used by MOVES Core Model



The **SHO**, **Starts**, **ExtendedIdleHours**, **OperatingModeDistribution**, and **SourceBinDistribution** CMIT tables have already been discussed. They are produced by various Generators and are read by the Core Model.

FuelSubtype is a category table, which establishes the fuelSubtypeIDs used in the MOVES Database. Every FuelSubtype belongs to a FuelType. It is also an information table that contains data items pertaining to these fuelSubtypes.

MonthGroupOfAnyYear is a category table that establishes the monthGroupIDs used in the MOVESDatabase. EPA originally envisioned that groups of months (or seasons) would be used along with calendar years to characterize the time periods pertaining to the fuel supply. However, in the default database distributed with MOVES2004, month groups are the same as the individual months, because the fuel seasons are not sufficiently uniform across the nation.

County is a category table that establishes the countyIDs used in the MOVES Database. Each County belongs to a State, but countyIDs are unique across the nation. This table is described in this section because fuel supply information is stored at the county level in MOVES2004. Its "altitude" field is not used in MOVES2004.

FuelSupply is an information table containing data about the market share of FuelSubtypes within FuelType for each county in each historical year and month. These marketshare values should sum to unity, so they are in effect a distribution. If a distribution is not present in this table, then MOVES2004 assumes that the gasoline fuel supply consists entirely of "conventional" gasoline and that the diesel fuel supply consists entirely of "conventional" diesel fuel.

FuelAdjustment is an information table containing multiplicative "fuelAdjustmentFactors" which are applied by the core model to the emission results for a pollutant process, source type, and fuelSubtype. A value of 1.0 represents no adjustment.

IMOBDAdjustment is an information table containing multiplicative adjustment factors which are applied to the energy consumption emission results to simulate the effects OBD type IM inspection programs. If no value is found in this table a value of 1.0 (no effect) is assumed.

SourceTypeModelYear was described previously in the context of the SBDG. It also functions as an information type table in the core model. It contains information as to what fraction of vehicles are equipped with air conditioning.

SCCVtypeDistribution is an information type table which contains distribution information for each combination of sourceTypeID, modelYearID, and fuelTypeID into the 12 vehicle types involved in the SCC category scheme. It is used by the core model to produce output broken down by SCC.

SCCVtype is a category table which defines the 12 "SCCVtypeIDs" involved in the SCC vehicle classification scheme.

SCC is a category table which defines the 144 SCC codes used to classify highway vehicles. Each of these is a combination of one of the 12 HPMS road types with one of the 12 SCCVtypes.

ZoneMonthHour is an information table which contains temperature and humidity information. A Meteorology Generator within MOVES2004 calculates the heatIndex values in this table.

FullACAdjustment is an information table which contains ACAdjustment values.

SourceTypeAge is an information table also discussed in the TAG section above. Its AC Fraction field is used by the Core Model.

EmissionRate is the principal information table used by the core model. It contains emission rates for pollutant processes other than Well-to-Pump.

DataSource is a supporting table, not actually used by the model that identifies the method used to derive each EmissionRate record.

GREETWellToPump is an information table that contains the emission rates for the Well-to-pump process. A default set of values is supplied with the model. The GREET model may be used to update these rates, by running a GREET simulation as a "preprocessing" step, and generating an alternate input table with the same schema as GREETWellToPump.

TemperatureAdjustment is an information table that contains multiplicative temperature adjustment values.

11.7.Miscellaneous Additional Tables

Finally there are several tables in the MOVES database which play minor roles or which are intended for use in future versions of MOVES:

State is a category table that defines the set of stateID values used in the database.

SCCProcess helps the model construct SCC code values and need not concern users of the model.

CountyYear associates all countyIDs with all yearIDs.

Grid, **GridZoneAssoc, LinkAverageSpeed, LinkHourVMTFraction**, and **GREETManfAndDisposal** are placeholders for future versions of MOVES, and are not used in MOVES2004.

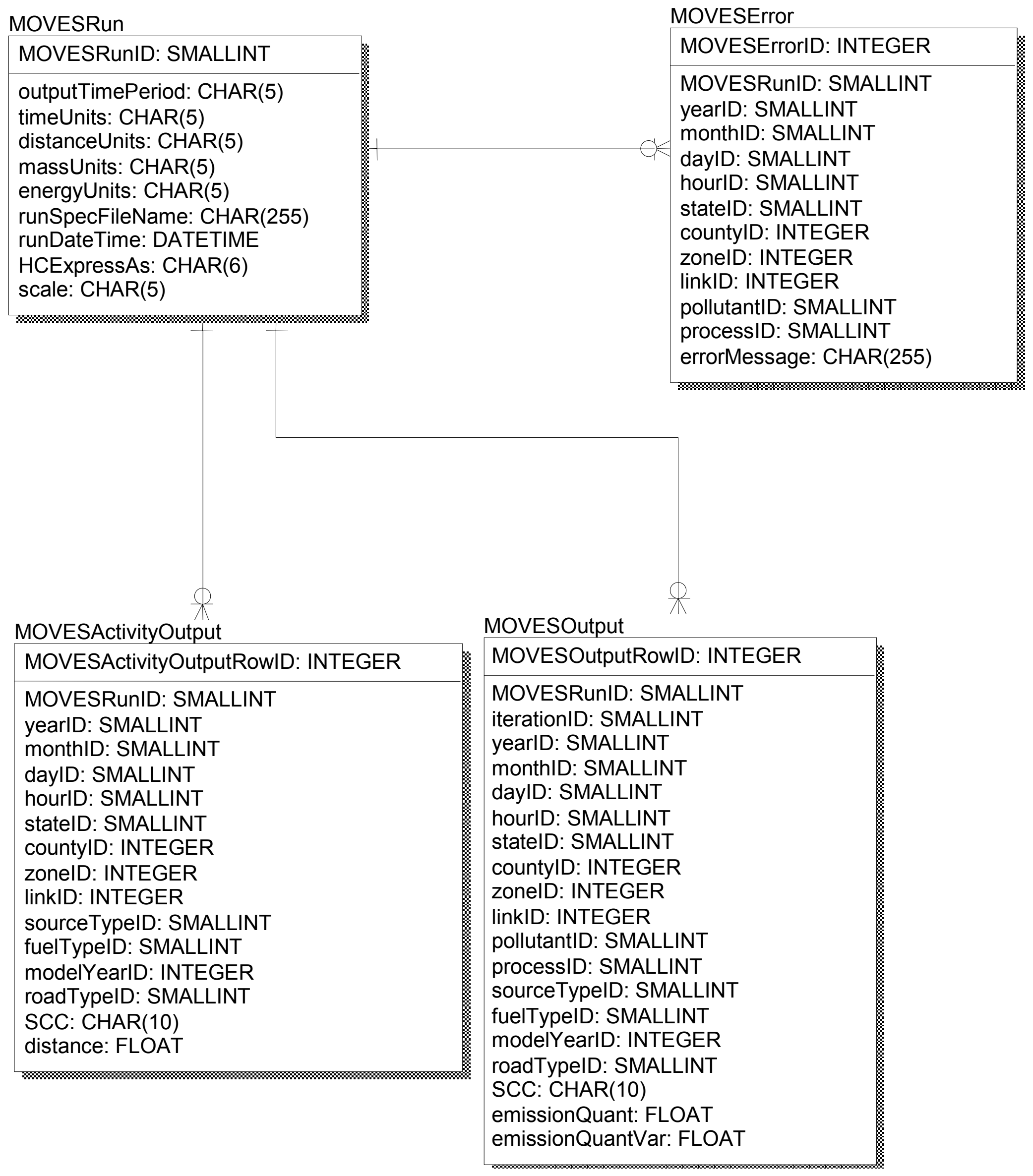
12. MOVES2004 Output Databases

The MOVES GUI/Master program reports the results of each simulation run in the MySQL database named in its run specification. The MOVES GUI may be used to create a new “empty” MOVES Output Database.

There are several software options for working with these databases once they have been created. The most natural and powerful method is to use MySQL itself, either via its command line client, which is installed as part of the MySQL installation, or via the MySQL Control Center which is a graphical client program for MySQL. The command line client is invoked from a DOS window by entering the “mysql” command. The MySQL Control Center requires a separate installation, which is included on the MOVES2004 installation disk. MySQL databases can also be used from Microsoft FoxPro or MicroSoft ACCESS via an ODBC connection. Even Microsoft Excel can be used in this fashion if the database is small enough. Detailed instructions as to how to establish these ODBC connections can be found in Appendix B of the *MOVES2004 User Guide.*

The results of multiple runs may be stored in the same database and are identified by MOVESRunID. This database consists of four tables as shown in Figure 12-1.

Figure 12-1. Tables in Output Database



12.1. MOVESRun Table

The MOVESRun Table contains information that pertains to the model run as a whole.

The MOVESRunID field contains a number identifying the model run. A single record is written into this table for each run. The first run whose results are stored in this database is assigned run number 1, and the number is incremented for any subsequent runs.

The outputTimePeriod field indicates the time period (Hour, Day, Month, or Year) for this run. This is a MOVES GUI selection.

The four Units fields, indicate the engineering units used to express time, distance, mass, and energy results for this run. These are MOVESGUI selections.

The runSpecFileName field contains the file name (not including path) of the RunSpecification upon which this run was based. Warning: the contents of this file may have changed since the run was performed.

The runDateTime field contains the date and time the run began.

The HCExpressAs field is not used in MOVES2004 and always contains NULL. (Its eventual purpose will be to indicate the hydrocarbon emission expression basis (e.g. total hydrocarbons (THC), volatile organic compounds (VOC), etc.) once estimation of gaseous hydrocarbon emissions is added to the model.

The scale field indicates the modeling scale applicable to the run. In MOVES2004 this field always contains the value "MACRO."

12.2. MOVESError Table

This table contains a record for each error message generated during runs for which this is the output database. Internally MOVES2004 has several levels of error messages. Currently "warning-level" and "error level" messages are written to this file, and "informational level" messages are not.

The MOVESErrorID field identifies the error message record. It serves as a key field for this table, but its value is not meaningful to the user.

As in the other MOVES Output Database tables, the MOVESRunID field identifies the model run during which the error occurred.

The errorMessage field contains the text of the error or warning level message.

The other fields in this table were intended to identify the portion of the model run that experienced the error, but are seldom if ever used.

12.3. The MOVESActivityOutput and MOVESOutput Tables

These two tables contain the substantive results of each model run. Both tables are described in this section since they have many fields in common. The MOVESActivityOutput table is used to report activity-related information which is not specific to an emission process or pollutant. Distance traveled is the only such result produced by MOVES2004. (Future versions of the model may add fields to the

MOVESActivityOutput table to report items such as number of starts, number of extended idle hours, etc.) The MOVESOutputTable is used to report the pollutant emission results, including energy consumption. The fields of these tables function as follows:

The MOVESActivityOutputRowID and MOVESOutputRowId fields serve to uniquely identify rows in each table, but their values are not meaningful to users.

The MOVESRunID identifies the model run that produced each output record.

The iterationID field (present only in MOVESOutput) is not used in the initial version of MOVES2004 and always contains the value of 1. It is intended to support estimation of the uncertainty of model results via Monte Carlo statistical methods in future versions. EPA's plans to add uncertainty to this model are discussed in section 9.18.

The yearID field identifies the calendar year to which the output record pertains. The Year table in the MOVESDefault database defines its legal values. The distributed version of MOVESDefault is intended to support calendar years 1999 thru 2050.

The monthID field identifies the month of the year (if any) to which the output record pertains. Its legal values are defined by the MonthOfAnyYear table in the MOVESDefault database and are 1 thru 12 in the distributed version. A null or zero value indicates that the record pertains to all months of the year that were included in the run specification.

The dayID field identifies the day of the week (if any) to which the output record pertains. Its legal values are defined by the DayOfAnyWeek table in the MOVESDefault database and are 1 (Monday) thru 7 (Sunday) in the distributed version. A null or zero value indicates that the record pertains to all days of the week that were included in the run specification.

The hourID field identifies the hour of the day (if any) to which the output record pertains. Its legal values are defined by the HourOfAnyDay table in the MOVESDefault database and are 1 thru 24 in the distributed version (where hour number 1 begins at midnight). A null or zero value indicates that the record pertains to all hours of the day that were included in the run specification.

The stateID field identifies the state (if any) to which the output record pertains. Its legal values are defined by the State table in the MOVESDefault database and are based on the FIPS state codes in the distributed version. A null or zero value indicates that the record pertains to all states in the modeling domain (normally the nation) that were included in the run specification.

The countyID field identifies the county to which the output record pertains. Its legal values are defined by the County table in the MOVESDefault database and are based on the FIPS state and FIPS county codes in the distributed version. Note that the county identifications do not rely upon the State table to be unique across the modeling domain. A null or zero value indicates that the record pertains to all counties in the state, (or if state is also zero or null the entire modeling domain) that were included in the run specification. When certain computational shortcuts are taken, as described in the GUI help screens, values of CountyID based only on the FIPS states codes are also used to represent entire states.

The zoneID field is based on the countyID in MOVES2004 and provides no additional information.

The linkID field identifies the link (if any) to which the output record pertains. Its legal values are defined by the Link table in the MOVESDefault database and are based on the FIPS state and FIPS county codes and road type classifications in the distributed version. A null or zero value indicates that the record pertains to all links in the county, that were included in the run specification. (In MOVES2004 this corresponds to all road types in the county that were included in the run specification.)

The sourceTypeID field numerically identifies the source use type (if any) to which the output record pertains. Its legal values are defined by the SourceUseType table in the MOVESDefault database. In the distributed default version these are:

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

A null value indicates that the output record does not pertain to a particular SourceUseType. (Either the user has selected to report by Source Classification Code (SCC) instead, or not to distinguish the results by any vehicle classification.)

The SCC field identifies the Source Classification Code (if any) to which the output record pertains. Its legal values are defined by the SCC table in the MOVESDefault database. A null value indicates that the output record does not pertain to a particular SCC. (Either the user has selected to report by Source Use type instead, which is recommended, or not to distinguish the results by any vehicle classification.)

The fuelTypeID field numerically identifies the top-level fuel type (if any) to which the output record pertains. A null value indicates that the record pertains to all fuel types. Whether results are to be distinguished by fuel type is a GUI selection and is included in the run specification. The legal values of fuelTypeID are defined by the FuelType table in the MOVESDefault database. In the distributed default version these are:

1	Gasoline
2	Diesel Fuel
3	Compressed Natural Gas (CNG)
4	Liquid Propane Gas (LPG)
5	Ethanol (E85 or E95)
6	Methanol (M85 or M95)
7	Gaseous Hydrogen
8	Liquid Hydrogen
9	Electricity

The modelYearID field identifies the model year (if any) to which the output record pertains. A null value indicates that the record pertains to all model years. Whether results are to be distinguished by model year is a GUI selection and is included in the run specification.

The roadTypeID field identifies the road type (if any) to which the output record pertains. The legal values of this field are defined by the RoadType table in the MOVESDefault database. In the distributed version there are 12 roadtypes corresponding to the 12 HPMS roadway classifications, plus a thirteenth RoadType representing off network locations.

1	Off-Network
11	Rural Interstate
13	Rural Principal Arterial
15	Rural Minor Arterial
17	Rural Major Collector
19	Rural Minor Collector
21	Rural Local
23	Urban Interstate
25	Urban Freeway/Expressway
27	Urban Principal Arterial
29	Urban Minor Arterial
31	Urban Collector
33	Urban Local

MOVES2004 associates start, extended idle, and well-to-pump emissions with RoadType 1 and running emissions with the 12 HPMS roadway classifications. A null value of roadTypeID indicates that the results pertain to all roadway types. Whether to distinguish results by roadTypeID is a GUI selection that is included in the run specification. Outputting results by SCC implies that RoadTypes must be distinguished.

The pollutantID field numerically identifies the pollutant to which the output record pertains. It is present only in the MOVESOutput table. Results are always distinguished by pollutant. The Pollutant table in the MOVESDefault database defines the legal values of pollutantID. In the distributed default version they are as follows:

5	Methane (CH4)
6	Nitrous Oxide (N20)
90	Reserved for Atmospheric Carbon Dioxide (CO2) (not operative in initial version
91	Total Energy Consumption
92	Petroleum Energy Consumption
93	Fossil Energy Consumption
98	Reserved for CO2 Equivalent (Not operative in initial version).

The processID field numerically identifies the emission process (if any) to which the output record pertains. It is present only in the MOVESOutputTable. The EmissionProcess table in the MOVESDefault database defines the legal values for processID. In the distributed version these are:

1	Running Exhaust
2	Start Exhaust
90	Extended Idle Exhaust
99	Well-to-Pump

A null value in this field indicates that the result record pertains to all emission processes. Whether to distinguish results by emission process is a GUI selection that is contained in the run specification.

The emissionQuant field is present only in the MOVESOutput table and contains the quantity of emissions of the given pollutant as qualified by all the other identifying fields. Engineering units for mass type pollutants may be in terms of kilograms, grams, pounds, or U.S. tons as selected in the GUI, contained in the run specification and output in the MOVESRun table. Engineering units for energy consumption results may be in terms of Joules or millions of BTU's as selected in the GUI, contained in the run specification, and output in the MOVERun table.

The emissionQuantVar field is present only in the MOVESOutput table and is not used in the initial release of MOVES2004 and always contains a zero value. It is intended for use in future versions of the model to contain an uncertainty estimate of the emissionQuant field value expressed as a variance.

The distance field is present only in the MOVESActivityOutput table and contains the distance traveled as qualified by all the other identifying fields. Its engineering units may be miles or kilometers as selected in the GUI, contained in the run specification,and output in the MOVESRun table.

Appendix I. EPA Response to Stakeholder and Peer Review Comments on *Draft Design and Implementation Plan for MOVES* pertaining to the MOVES Functional Design

In the fall of 2002 EPA published the pre-cursor to this document, "Draft Design and Implementation Plan for EPA's Multi-Scale Motor Vehicle & Equipment Emission System (MOVES)"". This report subsequently underwent two paths of review: public review, in which comments were solicited from model stakeholders, and independent formal peer review, in which comments were solicited from peer reviewers paid by the Agency according to Agency peer review guidelines.

This appendix provides a summary of those written comments received as a result of both review paths that pertain to the functional design of MOVES . The commenter of a specific comment is identified in parentheses, identified according to the bolded designation in the list below. Each comment number captures a unique sentiment, although variation on the general idea may have been submitted by several commenters as reflected in commenter identifications.

Comments were received from the following parties:

Public Review:

- Alliance of Automobile Manufacturers (**AAM**)
  - Review prepared by Tom Darlington, AIR
- Engine Manufacturer's Association (**EMA**)
  - Review prepared by Tom Darlington, AIR
- David **Roden**, AECOM Consult
  - Review prepared for U.S. DOT with respect to MOVES design applicability to TRANSIMS
- **U.S. DOT** – FHWA
- Peter **McClintock**, Applied Analysis
  - written comments in response to 11/2002 workshop
- Wayne Elson, **EPA Region 10**
- Donald **Stedman**, Professor, University of Denver
- **Natural Propane Gas Association / Propane Vehicle Council**
- Phyllis Jones, **North Carolina DAQ**

Formal Peer Review:

- Administered by Southwest Research Institute
- Reviewers:
  - Marc **Ross**, Professor Emeritus, University of Michigan
  - Ted **Russell**, Professor, Georgia Tech

- Michael **Replogle**, Transportation Director, Environmental Defense

Comments related to the software design are discussed below.

Scope of the Design

1. **Comment:** MOVES design is "remarkably flexible" and would be relatively easy to implement in many of the wide range of use cases (Ross). The multi-scale emission estimation architecture appears sound (Replogle). The model design is well thought out, provides a sufficiently broad range of use cases, and satisfies these use cases (Russell, McClintock). MOVES is highly compatible with the level of detail and focus of TRANSIMS (Roden).

   EPA Response: None required.

2. **Comment**: additional capabilities with regard to design are desired, as follows:
   a. Environmental justice evaluation (Replogle)
   b. Greater accommodation of use cases related to health effects and toxicity studies, e.g. modeling on 100 meter road segments or lane-specific information (Russell)
   c. Data visualization capability (Russell)
   d. Sketch modeling applications for transportation planning and traffic impacts, in collaboration with US DOT. Without this MOVES will be overly focused on policies related to vehicle technology and fuels (Replogle)
   e. "On-Road Microscale Inventory" scale to more explicitly accommodate TRANSIMS (Roden)
   f. Should explicitly include initial idle warmup emissions ad the effects of engine block heaters for cold climates (EPA Region 10, U.S. DOT)

   EPA Response: EPA feels that the MOVES design accommodates adding these capabilities, and certainly does not preclude any of them. MOVES output is broken down by geographic area, vehicle type, model year, vehicle age, etc. which could facilitate environmental justice evaluation. Its geographic links could be short road segments assuming that the user can supply information on that level. MOVES is designed functionally to accept the kinds of information produced by travel demand models, e.g. average speeds and link-specific vmt. Warmup emissions could be an emission process analogous to starting or extended idling implemented in MOVES2004. Actual implementation of these capabilities, however, which could be done by EPA or by other organizations, will depend upon demonstrated demand for them and some are bound to have higher priority than others.

3. **Comment:** model evaluation should be considered separately from model validation under the "model analysis" use case (Russell)

EPA Response: Our primary focus has been on model validation, e.g. comparison to top-down fuel sales estimates, but expect that evaluation will become more of a focus as the model begins to be applied to specific use cases.

4. **Comment:** Not enough information is provided to know whether MOVES will interact with SMOKE adequately (Russell).

   EPA Response: While this capability is not implemented in MOVES2004 the design does anticipate interaction with SMOKE by providing the ability to produce emissions as the level of user-defined geographic grid cells. The model also supports post-processing scripts, which we would envision would be necessary to make the link with SMOKE. We expect to expand on this "placeholder" as we move ahead with the criteria pollutant version of the model.

5. **Comment:** it would be useful to have MOVES link to transportation modeling programs such as TransCAD, EMME2, TP+ (Replogle) and PAVE (Russell).

   EPA Response: EPA agrees, and has designed the model to enable this. EPA and DOT are considering a collaborative effort to ensure the MOVES design adequately considers the range of travel models available. We feel travel model linkage can be accomplished by the addition of data importers to MOVES. EPA is hopeful that other organizations will help produce importers for specific products.

6. **Comment:** Specific attention should be paid to the variety of use cases associated with modifying vehicle operation, such as traffic calming, mode choice, and congestion pricing (Replogle)

   EPA Response: EPA feels the design facilitates these types of analyis, e.g. by the addition of internal or external "control strategies". The specific use cases mentioned in this comment likely wouldn't be internal to MOVES, but would rely on outside tools to estimate the effect of such strategies on VMT, which would then be fed to MOVES.

7. **Comment:** More detail is necessary on how the design accounts for two important sources of emissions, cold starts and high emitters (Ross)

   EPA Response: Start emissions are included in MOVES2004 as a separate emission process, and will continue to be refined with future versions. High emitters are more pertinent to criteria pollutant emissions, and will be addressed as these are added to the model.

8. **Comment:** "low-end" users are adequately served; concern is with high-end users. Will MOVES allow for sub-link analysis, e.g. spatial scale below a link (50-100 meters) in a convenient way? (Russell)

EPA Response: Yes; links can be defined in any way the user chooses, and could correspond to short road segments as long as the user can supply the information at that level.

9. **Comment:** What provisions are being designed in to identify errors and do quality assurance? (Russell)

   EPA Response: Most of these provisions are outside of the model itself and admittedly are not discussed in this design document. (The production of MOVES is being undertaken pursuant to a written Quality Assurance Plan, which is an internal EPA document.) In the development of MOVES, quality assurance procedures have taken place on many levels. With the software itself, the delivered product includes a set of Java class level unit tests based on JUnit, as well as a database consistency checking MySQL program that can be run on the MOVES default database and user input databases. EPA has also produced an automated suite of end-to-end tests that verify that over 100 specific variations in the input data have the expected effects on the results. The calculations have also been checked against separate "benchmark" calculations. Code updates are managed with the Concurrent Versions System (CVS) tool. Java source code is produced subject to a coding standard which includes Javadoc code commenting standards. The energy and emission data inputs have undergone additional quality checking, which are documented in "MOVES2004 Energy and Emission Inputs"

Total Activity and Operating Mode Formulations

10. **Comment:** With regard to the mathematical formulation presented in the design plan, modeling capability would be improved by using explicit models to produce what are proposed as static inputs, such as VMT, vehicle population, vehicle speed, temporal allocation of VMT and source hours parked. In general a significant investment in transportation and activity models, including TRANSIMS - by EPA, DOT and other organizations - is needed to make full use of MOVES (Replogle).

    EPA Response: We agree with this in concept, and perhaps in future versions the scope of MOVES can increase to include these suggestions via the control strategy design feature. For now, however, we envision that MOVES could work in conjunction with external tools design to address these issues, as discussed in Comment 6. The potential that MOVES provides in terms of more refined transportation and activity estimates is very large.

11. **Comment:** Whether mileage accumulation rates vary with vehicle age warrants some research (Replogle)

EPA Response: MOVES does allow for differences in mileage accumulation by age to be accounted for, and this is reflect in the default inputs (more details are available in "MOVES2004 Highway Vehicle Population and Activity Data")

12. **Comment:** Concern that the method proposed for determining VSP bins over 12 HPMS roadway types using average speed will fail to capture growth in recent years of higher traffic speeds, the shift from lower speed arterials to higher speed freeways, and the potential for further shifts in speed distribution within roadway types (Replogle).

    EPA Response: The VSP bin approach does not preclude accounting for these issue. If inputs of average speed and roadway VMT distribution are available to account for these shifts, MOVES will be able to account for them.

13. **Comment:** The user's ability to gather data on total activity should be relatively straightforward, but not so for operating modes. Operating modes may vary by region, calendar year and vehicle age, which would comprise the use of generic data derived from driving survey data (Ross). The distribution of activity by VSP bin should be validated using instrumented vehicle data (AAM, EMA)

    EPA Response: MOVES2004 will rely on default operating mode distributions derived from an estimate of national average survey data; as we continue to collect in-use driving survey data, these estimates will be refined.

14. **Comment**: The issue of aggregation across scales requires close attention to network specifications in mesoscale models and understanding the limitations of HPMS. This requires more attention from EPA and should be addressed as part of a larger strategy to integrate MOVES with TRANSIMS and other tools (Replogle). Should detail limitations of HPMS (U.S. DOT). Using the same modeling approach on all three scales could result in significant problems with the macroscale inventory (AAM, EMA).

    EPA Response: We acknowledge that MOVES20o4 would produce different results for the same area modeled at different scales. One reason for this is that, as mentioned in the comments, the activity data sources used at differing scales (i.e. HPMS and travel models) are not perfectly consistent. Another reason is that the difference between the scales comes down to aggregration levels, and different levels of input aggregation inherently lead to different results (this issue is not unique to MOVES). We expect that policy guidance will be needed to determine the appropriate scales to employ depending on the application and available data.

15. **Comment:** Since SHO is the key activity variable, the approach calculating SHO from speed and VMT should be verified (AAM, EMA). Concern with vehicle mileage/hours and speed and the degree to which these data will be produced satisfactorily especially for mesoscale operation (Replogle).

EPA Response: SHO, speed and VMT are linked by the relationship SHO = VMT/speed. Many modeling frameworks, including MOBILE, use VMT and speed in conjunction with one another; the MOVES approach is another variation on this. In terms of mesoscale, we expect inputs to be generated by travel models which work with all three variables in producing link-based average speed.

16. **Comment:** In order to not "seriously shortchange" an important group of users, ensure that good and easy-to-use default information, or data generation capability, be available to relatively modest users at the mesoscale (Ross). Some users may be disheartened by the potential data requirements, but the establishment of national default data to fill some gaps addresses this concern (Replogle)

    EPA Response: While our hope is to minimize the burden on finer-scale users, some data input will be required at these levels. Many of the default level inputs developed in MOVES2004 for macroscale can be applied to the mesoscale and microscale as well, with the exception of link-level speed and VMT. These inputs are location-specific, and we are not planning on developing default link-level inputs for the entire nation.

17. **Comment**: More information is needed on how users would supply local travel activity data and road grade, and how it would be translated into a useable form in the model (U.S. DOT). Assuming a default grade of zero would be problematic (Replogle)

    EPA Response: Grade comes into the model as a term in the vehicle specific power (VSP) equation, and can be entered into the model at the link level (with users defining what a link is). As a default case for MOVES2004 it is assumed to be zero – we have yet to work out a satisfactory method to account for grade at the macroscale. Even for mesoscale or microscale applications reliable grade data might be hard to find, or may require extensive processing of GIS and/or elevation data to develop link level grade inputs for MOVES. MOBILE has long been criticized for not accounting for road grade; MOVES had therefore been designed to include it if available, but this doesn't make the data any easier to find. Zero grade can always be assumed.

18. **Comment**: Does EPA intend to develop new driving cycles representing intersection activity? (U.S. DOT)

    EPA Response: MOVES2004 includes new driving cycles for medium and heavy-duty vehicles, and some new cycles for high-speed light-duty operation. Intersection activity is inherently accounted for in the non-freeway cycles for all vehicles, but new cycles specifically representing intersection activity have not been developed. This could be considered if this was an important priority for MOVES users.

19. **Comment:** Does EPA intend to develop a companion intersection dispersion model to replace CAL3QHC, or re-validate CAL3 using MOVES? (U.S. DOT)

    EPA Response: The development of dispersion models is outside the scope of MOVES and is not part of the MOVES development plan, but when microscale applications are addressed we will investigate the latest developments in dispersion models and consider how MOVES can work with them.

County Database

20. **Comment:** Concept of a "County Default Database" is supported, including emissions. Additional breakdowns may be required in Alaska; Tribal Default Database should also be considered (EPA Region 10)

    EPA Response: This is possible in the MOVES design framework, but would require additional consultation with stakeholders to move ahead with.

Input/Output/Model Operation

21. **Comment:** MOVES should allow more sophisticated users to bypass the GUI (Russell)

    EPA Response: MOVES2004 does include a Command Line Interface which allows the user to completely bypass the GUI. Run specification information is stored in an XML format which can be modified (or even produced) without using the GUI.

22. **Comment**: It would be useful to archive calculations for efficiency, e.g. location-specific I/M results (Russell)

    EPA Response: We would have to understand better what calculations users would be interested in seeing, and in what format.

23. **Comment:** MOVES should be designed to run on operating systems other than Windows, e.g. LINUX (Russell)

    EPA Response: MOVES is designed to be platform independent. Its generic functional design is implemented with the Java programming language and MySQL and requires only shared file services to operate over a computer network. These tools were selected in part because of their high degree of platform independence. The Java programming language is designed to execute on any platform for which a 'Java Virtual Machine' is available which includes UNIX, LINUX, Macintosh, etc. The MySQL database management software is written in C++ and is also available on many platforms including UNIX and

LINUX. EPA has only implemented and tested MOVES2004 on the WINDOWS platform, but it should be feasible for others to port MOVES to other platforms if desired.

24. **Comment:** Include a fast-run feature that requires few if any inputs and bypasses the standard screens (EPA Region 10)

    EPA Response: The MOVES Graphical User Interface does not require the user to visit any particular screens. An existing run specification can be loaded, run as is, or run with as few changes as desired. The MOVES Command Line interface can also be used to run MOVES directly from the command line, or via another computer program, without using the Graphical User Interface at all.

25. **Comment:** Results should be available in various forms to help understand results, such as ratios of emissions to fuel or by model year group (Ross)

    EPA Response: Because so many different result forms are likely to be needed, MOVES produces its "raw" results a single, very general database form. These raw results can be very detailed, or, for considerations of data size and model performance, they can already have been aggregated by the model to various levels. (Over 100 variations are possible.) An easily extensible set of post - processing MySQL scripts, several of which are included in MOVES2004, can then be used to summarize the results in whatever particular way is desired. If the set of result information is small enough one of these provides for the results to be exported to a spreadsheet format.

Appendix II. Pre-Publication Peer Review Comments and EPA Response

An earlier draft version of this MOVES2004 Software Design and Reference Manual, dated September 8, 2004, was reviewed by Armistead Russell, the Georgia Power Professor of Environmental Engineering of the Georgia Institute of Technology's School of Civil and Environmental Engineering. This review was conducted pursuant to EPA peer review process. Changes made to produce this version, which is still a draft, have been made primarily in response to this review. Professor Russell's comments, interspersed with EPA's responses, form the remainder of this Appendix.

Review of MOVES2004 Software Design Document (and initial implementation of the MOVES2004 Software)

Armistead G. Russell

MOVES2004 is the first substantiation of EPA's next generation mobile source emissions inventory system, limited currently to applications involving fuel use and greenhouse gas generation. Capabilities for providing emissions of pollutants important to other air quality issues are forthcoming. This review of the present system is from the perspective of a principle investigator (PI) leading a group who will likely use the software for studying greenhouse gas, particulate matter (PM), ozone and carbon monoxide (CO) issues. A major thrust of our work is to assess how future control strategies will impact air quality and uncertainties associated with the air quality modeling process. Ultimate applications of the research are typically in the science, health and policy areas (roughly in that order). We currently use MOBILE in our applications.

In conducting this review, I first read through the Software Design and Reference Manual (SDRM) document for content, then loaded and "tested" the software, and then went through the SDRM to make editorial suggestions/corrections and further understanding. The final step of the review was drafting this document. As initial comment, the "testing" of the software was limited to (attempted and successful) installation and running of two cases. This review does not constitute a test of the software, itself.

As an immediate impression of the SDRM, at times it was, a relatively easy read providing a solid feel for the software and its design. At other times, the SDRM was awkward and much more difficult to parse, greatly slowing down the review process.

There were inconsistencies in the use of names (capitalization, spaces, etc.). Fonts changed (which may sound like a trite problem, but is not the case with such a document, as it can lead to issues when a 1 or l (a “one” or “el”), or o or 0 is used. (In one font that was used, a zero looked like an “oh.”) In some of the mathematical equations, the change of style was bothersome as it conveyed that there might be a different meaning or importance. These are issues that need to be dealt with in the next round.

EPA Response: Most of these inconsistencies in the capitalization and spacing of names, and font changes have been corrected. A section explaining the naming scheme the document tries to use has been added at the beginning of the document.

The SDRM describes a system that appears to have significant flexibility for future use for mobile source inventory development. The system breaks down the process into relatively small pieces of the calculation, such that as the practice of mobile source modeling evolves, the software is capable of evolving as well, and will be able to capture those changes with relatively small amounts of effort. The software is based upon MySQL, which enjoys a wide availability and use. This avoids some of the current problems with MOBILE, which is based on a more primitive language. Though this brings up a performance issue. My view is that performance is not a key problem at this point for two primary reasons. First, as designed, MOVES can be applied on multiple processors (though I did not test this feature, instead only applying it in the single processor mode). Second, processor speed will continue to improve, reducing those issues. On the contrary, it does impact the ability to do detailed uncertainty analysis, particularly given when the approach to uncertainty analysis is being planned (more on that issue later). We do not find running MOBILE to be a major issue from a computational speed viewpoint, but it is an issue in terms of ease of use, particularly when doing something non-standard.

EPA response: model performance had been improved in the version loaned to this reviewer relative to earlier versions used internally by EPA and for “beta testing”. EPA still feels that further improvements would be highly desirable, but further improvement is not apt to be possible for the initial release of MOVES2004.

A few structural improvements in the SDRM are in order. First, the title page should have the report authors, along with any other necessary EPA contact information. The software designers should be directly acknowledged either on the title page or inside near the front. Next, the SDRM needs an abstract to give the reader an approximate view of what is to come. I would envision two to three pages. This should provide a top-level description of the system and what is (and is not) contained in the SDRM. Third, the SDRM could benefit greatly from a table of acronyms. This will, make reading the document more straightforward, and will make it so you are not left up in the air when you see an acronym in a table, when that acronym has yet to be defined in the text (also noted in my edited version are times to add acronyms to the tables, themselves). A final structural addition would be a text cut-out in the Introduction that describes the naming convention used, when capitalization, blanks, or concatenated words are used, when bold or italicizing are used, etc. This will help not only the reader, but the authors working through the manuscript as well.

EPA Response: The title page has been expanded to list the report authors and other software designers Contact information will be given on the web page from which the document is downloaded. While not producing a separate abstract, the document introduction has been rewritten and expanded to provide the information suggested. Another document, the "Road Map to MOVES2004" will be produced explaining the role of each MOVES2004 document including this one. A table of acronyms has been added as Appendix III. A section has been added immediately following the title page explaining the naming conventions used.

In the SDRM, it would be beneficial to include a cut-out box to discuss the use of MySQL vs. Access or other database system, what ODBC is and does, etc. This would be written for the user with some familiarity with Access (or another system to help them decide what would be involved, and how to proceed further. It might also help newer users understand why one might want to go to the next step. For example, we use Access in our emission inventory processing, and if what would be involved in linking MOVES2004 with access, was better understood, linking MOVES2004 with Access might be my choice.

EPA Response: A section explaining how to use Microsoft ACCESS via Open Database Connectivity (ODBC) has been added to the MOVES2004 User Guide and a reference to the relevant section of that document added.

The SDRM does not provide enough information about how it will link with SMOKE and PAVE. Given the widespread and growing use of SMOKE, this is a shortcoming. Even if this is a feature that is planned, but not yet there, a section discussing this future capability is needed. I realize that this is not so important for the first substantiation of MOVES for greenhouse gases, but it would still have uses in that regard. For one, if one were to use fuel-based emissions inventorying procedures to estimate VOCs, CO and NOx, this might be of interest. (Though, that brings up a whole new set of issues.) Also, if it works well with SMOKE, then that can ease its use with PAVE. Likewise, PAVE is widely used in the air quality community, and the ability to directly use PAVE with MOVES2004, or at least have the SDRM discuss how this might be done. In the future, this should be a supported capability.

EPA Response: This is indeed a feature that is "planned, but not yet included". Consideration has been given in the design to how grid cells are related to the geographic entities in MOVES We do not think that interfacing MOVES to gridded air quality processing models such as SMOKE, will present major design difficulties.

<u>Overarching Concerns/Possible Limitations (that may be no problem, already captured, etc., but not fully described/covered by the SDRM)</u>

There were two major areas that, as noted in other areas of the review, may be limitations. First, it is not apparent how well the system captures the impact of hybrids. This, to me, is because the system currently uses a binning approach and VSP. The VSP-emissions/fuel use relationships were developed for conventional vehicles. It does not show up in the menu of fuels. It may never be a major player, but it is making a splash for now.

EPA Response: From a database perspective hybrids show up in the list of engine technologies, rather than in the list of fuels, since hybrid vehicles will use the same fuels as conventional vehicles. It was indeed unclear in the draft document reviewed how emission rates for hybrid technology vehicles are developed. A component has been added to the model, the Future Emission Rate Creator (FERC) which EPA had used (as a separate component outside MOVES2004) to produce a default set of rates for hybrid technology vehicles, and which model users can use to model alternatives. This component has been added to the data flow diagram and discussion in Chapter 8, and a functional specification for it added to Chapter 10. This component, along with the Alternative Vehicle Fuels and Technologies (AVFT) Strategy component, which generates source bin distributions for future technology vehicles, can be used to estimate the impact of hybrids.

A second concern is the ability to readily conduct sensitivity and uncertainty analysis. This is not part of the system, though it does have an iteration number that can be used when doing Monte Carlo (MC) analysis at a higher level. Apparently, one would develop a MC analysis routine, and exercise MOVES in individual calls as the parameters are varied. It is recommended to have the capability within MOVES since requiring users to build their own MC driver, develop the links, etc., might be beyond many individuals. One can then take more advantage of not having only information on the nominal data values, but uncertainties and sensitivities in the data structures as well. Further, it will likely reduce mistakes by others. Requiring the uncertainty information in the data structures will also lead to more attention being given, up front, to that need. Then, having that data will also facilitate the analyses. It is a win-win situation. There should also be an indication of the confidence one has in the uncertainty estimate. Being able to conduct the uncertainty analyses, as part of MOVES will require additional mathematical and computational capabilities, essentially paralleling the current MOVES steps. This would involve a traditional propagation of uncertainty calculation, assuming normality (which is probably fine for most analyses), and directly utilizing the sensitivities (which should also be propagated). Building in such a capability up front can make such a process more efficient than leaving this capability to be added at a later time. It can identify limitations in the current approach as well (though I have not noticed any).

EPA Response: Section 18 has been added to Chapter 10 of this document describing the attempts EPA has made to date to incorporate estimation of uncertainty into MOVES2004, including fundamental problems encountered using the propogation of error method, and EPA's plans to estimate uncertainty via a simple monte carlo approach in a subsequent version. This would involve adding a

looping level within the model as suggested. Uncertainties, but not sensitivities, would be stored in the database, the design of which already includes some of the fields necessary. Repeated use of the feature (which itself involves multiple model run iterations) would still be needed for the user to derive an "indication of the confidence one has in the uncertainty estimate." EPA expects that performance limitations will severely limit the scope of runs for which uncertainty estimation will be practical.

As part of the review, we were asked to address specific aspects of MOVES2004 and the SDRM. These are covered below.

Software Design

In general, I am pleased with the apparent software design. It uses advanced, yet widely available, used and evolving, approaches, e.g., using MySQL and a good GUI. The system is modular, and breaks the operations down to a fine level of detail. This should allow the system to evolve relatively easily, as long as there are no radical changes in the approach. As I have more tangential knowledge of how others have approached the design of mobile source emissions systems, I would recommend that individuals from groups who have designed and/or developed emissions inventory systems also review this aspect of the design (e.g., Barth from UC-Riverside, Guensler/Bachman from GIT, Pechan, Alpine Geophysics, Environ, etc.).

While it may be more complicated than it is worth, in my opinion it would be useful if there was a relatively easy way to integrate in other emissions modeling approaches, e.g., replacing the current approach. For example, how easy would it be to switch over to using California approaches? How about fuel-based approaches? This would be nice if for no other reason to facilitate comparison between the approaches. This may be viewed as a functionality issue, not a design issue.

EPA Response: Chapter 4 of another document, the MOVES User Guide, deals with "Customizing MOVES2004" and is intended to meet this need.

Intended Functionality

Recognizing that MOVES2004 is a first substantiation of a more general MOVES, and that it is to be applied for greenhouse gas emissions, and related quantities, it appears to provide a solid foundation in terms of functionality. It currently has, what I would view, are most of the major needs in terms of capturing the range of equipment, driving conditions, and impacts of alternative fuels. One shortcoming, which is from a lack of knowledge, is that it currently assumes roads are level, which, in turn, impacts VSP and therefore gas mileage and emissions. If nothing else, one should build in a factor to be used to account for roads with positive and negative grades. This will become an interesting component in regards to using hybrids and being able to capture their emissions (greenhouse and otherwise). (That gets back to the overarching concern discussed above: how do you currently treat hybrid technologies, e.g., with regenerative breaking, etc.?)

EPA Response: The "grade" field in the Link table, along with the fact that operating mode distributions and the MOVESOutput records, are distinguished by linkID, is designed to accommodate this. An earlier version of MOVES2004 even included the grade value in the VSP calculation, so we have actually tried this successfully. Grade was removed from the VSP calculation in this version of the model because it is not necessary at macroscale, and there was some penalty in terms of execution time performance. EPA plans to include the grade field value in mesoscale and microscale calculations when they are implemented, which is one reason why MOVES run specifications include an indication as to their scale.

Report Clarity

As noted in the beginning, the report is a bit schizophrenic. Parts are quite clear, while others are not. It suffers from lack of consistency in style, use of nomenclature, capitalization, concatenation, etc. There are areas that require more explanation. A particular need is for the installation and "Getting Started with MOVES2004" discussion to be fortified dramatically. The online "Users' Manual" may be the place to provide the most detailed discussion, but it is currently insufficient, and more in the SDRM would be useful, if for no other reason to let its readers know what is involved somewhat better than is currently the case.

EPA Response: We have improved the clarity and consistency of the document. "Getting Started with MOVES2004" is better dealt with in the MOVES2004 User Guide and the MOVES2004 Installation Guide. The introduction to this document has been rewritten to try to make this more clear. Another top level document, the Road Map to MOVES2004, will steer readers who are getting started in that direction.

A particular problem is that many sections start out with little transition from the prior section, and change abruptly. Each section should start out by putting the next paragraphs in perspective. There should be a common format in terms of lists (do you use bullets, numbering, lettering, etc.). There are some paragraphs made up of single sentences (not necessarily wrong, but often indicative of a problem) that come out of the blue, and end there, too. All of these problems are exacerbated by the changing fonts, bolds, capitalization, etc., used in the SDRM. I have indicated in my comments directly upon the SDRM manuscript what portions are most clear. These should serve as templates for the rest of the document.

EPA Response: Most of these items, including everywhere they were specifically identified in the marked up paper copy provided by the reviewer, have been improved. We have not gone to thelevel of effort that would be required to "put all lists in a common format."

Application to Use Cases

While the actual use cases suggested by EPA were not included in this document or transmittal letter, I suspect this refers to the "Use Cases" from the last review. As such, I saw no further limitations from the last review, so it would appear that it can capture those uses.

EPA Response: The reviewer is correct that the use cases referred to were those from the last review, namely those discussed in the Draft Design and Implementation Plan for MOVES published in October, 2002.

Does the design allow for a range of input data options?

From what I can tell, and this is rather out of my range of expertise, the system provides the ability to accommodate the needs of most users (with limitations discussed elsewhere, e.g., for conducting uncertainty analysis), and to do so without an undue burden. In this case, one has to caveat undue burden, in that it can, and likely will be, a significant burden, but not undue. Further, this burden will likely be shared amongst the community. On the other hand, I am not sure that the accuracy sacrifices noted as one goes to a higher level of aggregation are needed, and if such a sacrifice is necessary. In the design, would it be possible for the data to also capture how different levels of aggregation will impact emissions estimates. Would it not be possible that the fine scale information also contain information as to how to adjust their part of the emissions as one uses a higher level of aggregation, and that information is easily to the more aggregate calculation in a computationally efficient manner? This may involve using defaults, i.e., if that information does not exist; the multiplier is one for all pollutants. If it does exist, that information is passed up the ladder. For example EPA might provide the master data set with such information attached in the data files, which would be accurate if the default files are used. If one modifies the input data, they would need to re-run the system, using the least aggregated level, produce the needed information, and then populate the current data files with that information for future use.

EPA Response: EPA believes the accuracy sacrifices noted as one goes to higher levels of pre-aggregation are in general necessary, though some particular sources of inaccuracy could be removed by pre-aggregating the data in a more complex fashion. To the extent this is done, however, the calculations would take longer, and it is the purpose of data preaggregation to speed execution time. The inaccuracies introduced by the pre-aggregations as currently being performed are being characterized by EPA. Post aggregation which can also be specified in the MOVES GUI and run specifications does not involve these accuracy sacrifices.

Another concern in terms of the design and input data options relates to the ability to readily conduct sensitivity and uncertainty analysis. It would be useful if not only nominal values are available in the input data, but also an estimate of uncertainty and the sensitivity of the parameter to various input parameters. As noted in this review where uncertainty is directly discussed, the current design revolves around using a Monte Carlo approach, which is time consuming, not always necessary, and sometimes less informative than a more direct approach. Further, if the question is how sensitive the result is to a specific set of N inputs, it can be faster to not have to run N simulations. (Also, if non-linearities are involved, that would require a couple of extra simulations to get a centered-difference approximation of the sensitivity, though I doubt this is an issue).

EPA Response: Section 10.18 has been added to this document explaining how EPA plans to add a simple, straightforward monte carlo method of uncertainty estimation to the model. EPA expects, however, that execution time performance will limit the scope of its application. EPA is also analyzing the sensitivity of the model to variations in the input data and will publish these results, which EPA agrees are more useful for many purposes.

Are the algorithms a reasonable approach across multiple scales?

My review of the way the calculations are broken down suggests that the system can be very fine grained such that it can capture very fine spatial and temporal scales, IFF the data can support the calculation. Further, the calculations are broken down to very basic steps, such that if a more complex (or simpler) approach becomes warranted, the system is readily modified. While there is a performance penalty to do the operations at such a fine scale, as noted earlier, I do not believe that is a significant issue.

Will the proposed design of MOVES permit an accurate assessment of fuel economy and greenhouse gas emissions?

Here, my review suggests that the system will permit an accurate assessment of the desired quantities, keeping in mind that the real limit here is the underlying science. (One exception is the use of hybrids, as noted above. It is not apparent, to me, how well this technology is captured, since it is so different and it is not currently included in the list of technologies treated.)

EPA's response to this topic is given above.

Software Installation and Use

I initially attempted to load the MOVES software on one of my home computers, an Athlon-based machine running Windows XP Home Edition, with Service Pack 1. While all the files appeared to have unzipped correctly, the MySQL set-up would not run. Re-attempting the unzipping and loading was still unsuccessful. I have not yet determined the reason why. Next, I copied and unzipped the files on a second machine, also running XP (though the professional version). This time, loading MySQL and MOVES2004 went smoothly. The SDRM, and the online Users' Manual were adequate for the purposes of loading the software, at least for a single processor application.

EPA Response: It sounds like this problem might relate to the home edition of Windows XP, but we are unable to replicate it and are not aware of anyone else experiencing it. We have installed MySQL successfully on at least one XP home edition system. A quick review of MySQL AB's documentation of known problems

did not find anything specifically pertinent to the home edition of Windows XP. We'll keep an eye out to see if others experience similar problems as the model is more broadly distributed.

Next, I conducted a first application, using the instructions in the Users' Manual. While not part of this review, this took a bit long, primarily because the current Users' Manual is inadequate. I would recommend adding to both the Users' Manual and the SDRM a step-by-step set of instructions to run the default case (including what to expect when you single click individual functions, how to make changes, etc.), and the output to check the results. The current instructions point you in the correct direction, but don't quite give you adequate instruction to proceed efficiently and confidently.

EPA Response: Chapter 3 has been added to the MOVES2004 User Guide to give step-by-step instructions as to how to execute the example run specification.

<u>Summary</u>

The MOVES2004 SDRM is a reasonable initial document describing the software, and the software, as described is a solid first substantiation of the MOVES system. The SDRM is not clearly written throughout the document, though portions are in good shape. Weaknesses in the design appear to be in its ability to do direct sensitivity and uncertainty analysis, readily capture alternative approaches to estimating mobile source emissions (e.g., fuel-based, California, etc.), and it does not appear to have the ability to treat hybrid vehicles.

EPA Response: EPA appreciates the thorough review of this document performed by Professor Russell. The items mentioned in his summary have been responded to earlier in the document where they first appeared.

Appendix III. Table of Acronyms

AB	aktiebolog (Swedish)
AC	air conditioning
API	application program interface
AVFT	alternative vehicle fuels and technologies
ASCII	American Standard Code for Information Interchange
BTU	British thermal unit
CH4	methane
CD	compact disc
CMIT	core model input table
CNG	compressed natural gas
CSV	comma-separated variable
DOT	Department of Transportation
ECC	energy consumption calculator
EPA	Environmental Protection Agency
F	Fahrenheit
FERC	future emission rate calculator
FK	foreign key
FIPS	federal information processing standard
GB	gigabyte
GPL	general public license
GHz	gigahertz
GUI	graphical user interface
GNU	GNU's not UNIX (recursive)
GREET	Greenhouse gases, Regulated Emissions, and Energy uses in Transportation
H2	hydrogen
HC	hydocarbons
HDT	heavy duty truck
HPMS	Highway Performance Management System
IC	internal combustion
ID	identification
IM	inspection/maintenance
kW	kiloWatt
LDT	light duty truck
LDV	light duty vehicle
LPG	liquified propane gas
MAR	mileage accumulation rate
MB	megabyte
MOVES	MOtor Vehicle Emissions Simulator
N2O	nitrous oxide
NMIM	National Mobile Inventory Model
OBD	on board diagnostics
ODBC	open database connectivity

OMDG	operating mode distribution generator
OTAQ	Office of Transportation and Air Quality
PTW	pump-to-wheel
RAM	random access memory
RTC	runs to completion
SBDG	source bin distribution generator
SCC	source classification code
SDK	software development kit
SDRM	Software Design and Reference Manual
SHO	source hours operating
SQL	structured query language
SUV	sport utility vehicle
TAG	total activity generator
US	United States
VMT	vehicle miles traveled
VOC	volatile organic hydrocarbon
VSP	vehicle specific power
WTP	well-to-pump
XML	extended markup language