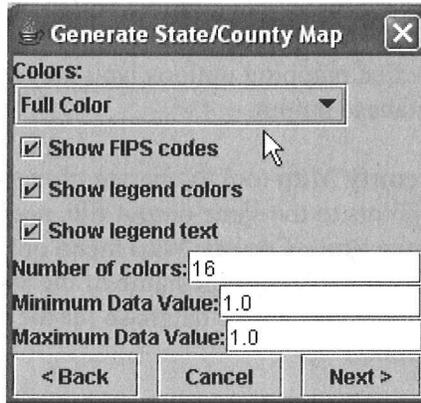


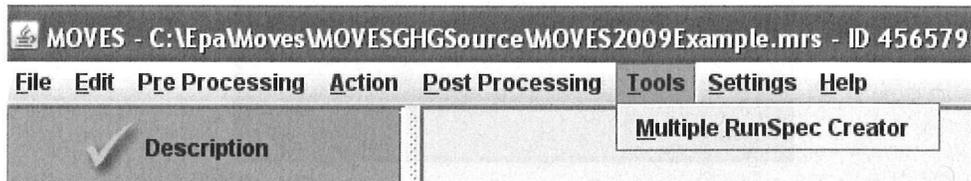
the user instructs the mapping tool to use results from the RunSpec to generate the map output. After making selections in this window, the user should click "Next".

A window allowing the user to define map attributes opens. The user can choose "Full Color" or "Grayscale" using the Colors drop-down box, and can customize the map using the check boxes and defining minimum and maximum data values for the emissions output data. When the user clicks "Next", the map output is generated.



### 2.3.6 Tools

Tools provides a drop-down menu with the choice **Multiple RunSpec Creator**.



This menu item contains a list of MOVES tools. In MOVES2010, the only tool on the list is the **Multiple RunSpec Creator**, which creates a set of new MOVES runspec files using the currently loaded runspec and a user-supplied control file as a template.

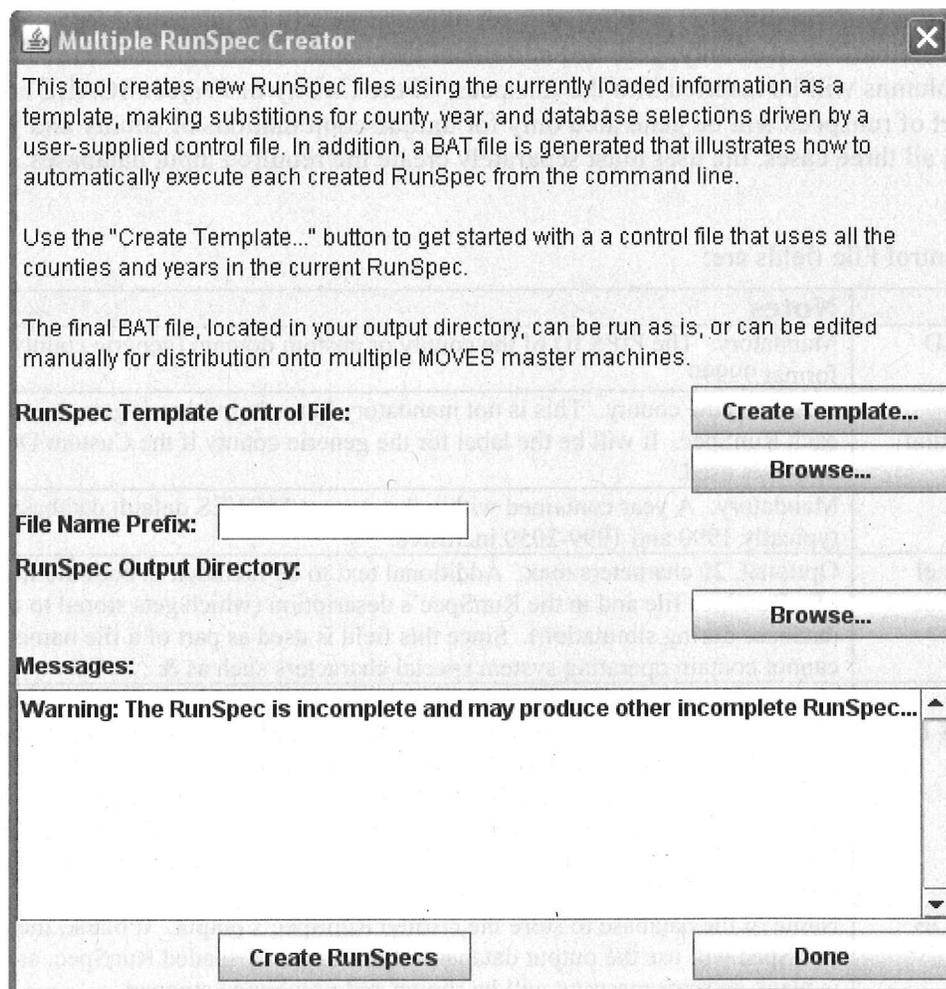
#### 2.3.6.1 Multiple RunSpec Creator

The **Multiple RunSpec Creator** creates a set of new MOVES RunSpec files using the currently loaded RunSpec information, and additional information in the form of a user-supplied control file. The control file is a list of county, year and database data selections in the form of a Microsoft Excel spreadsheet file (.xls). The tool produces a set of MOVES RunSpecs and a batch (.BAT file) file that can execute the MOVES RunSpecs from the DOS command line (MOVES batch files cannot be executed from inside the MOVES GUI).

Before using the **Multiple RunSpec Creator**, the user should load or create a viable (all green check marks) Runspec (the tool will accept an incomplete RunSpec, but will subsequently

generate a set of incomplete RunSpecs which require further manual editing). The **Multiple RunSpec Creator** also requires the selection of at least one county and one year in the RunSpec.

If the menu item **Multiple RunSpec Creator** is selected, the “**Multiple RunSpec Creator**” panel opens. The top of the panel contains some reminders about the purpose of the tool and mentions the output .BAT file. Error messages and other feedback to the user are provided in the messages area at the bottom of the window.



Once the panel is opened, use the “**Create Template...**” button to get started with a Control File that uses all the counties and years in the current RunSpec. The **Create Template** feature will fill the Control File with the information available in the RunSpec. However, the user may edit the Control File and add additional rows and columns of information. The user may add additional counties, years and database names. For each row in the Control File MOVES will create a new RunSpec.

The figure below shows a template Control File (an .XLS in this case) that will create RunSpecs for a list of counties, each with a different output database:

	A	B	C	D	E	F	G	H	I
1	CountyID	County Description	Year	Additional Text Name	Primary MOVES DB	Output DB	Domain DB	Advanced Features DB	User DB
2	26161	MICHIGAN - Washtenaw County	2009	Washtenaw	MOVESDB20090112	WashtenawOutput			
3	26163	MICHIGAN - Wayne County	2009	Wayne	MOVESDB20090112	WayneOutput			
4	26161	MICHIGAN - Washtenaw County	2010	Washtenaw	MOVESDB20090112	WashtenawOutput			
5	26163	MICHIGAN - Wayne County	2010	Wayne	MOVESDB20090112	WayneOutput			
6	26161	MICHIGAN - Washtenaw County	2011	Washtenaw	MOVESDB20090112	WashtenawOutput			
7	26163	MICHIGAN - Wayne County	2011	Wayne	MOVESDB20090112	WayneOutput			

**!Tip** If the *National* SCALE is selected, a set of runspecs will be generated for unique combinations of county, year and additional text name. Information from the other columns will be inserted into the runspecs. If the *County* or *Project* SCALE are selected, a set of runspecs will be generated only for unique combinations of county and year. Note, in all three cases, the user must separately create the required input databases.

The Control File fields are:

Field	Notes
CountyID	Mandatory. The FIPS ID of the county or custom domain (generic county) in the format 99999
County Description	Describes the county. This is not mandatory but is helpful as it gets stored within each RunSpec. It will be the label for the generic county if the Custom Domain option is used.
Year	Mandatory. A year contained within the current MOVES default database, typically 1990 and 1999-2050 inclusive.
Additional Text Name	Optional, 20 characters max. Additional text to be included in both the file name of a RunSpec file and in the RunSpec's description (which gets stored to the output database during simulation). Since this field is used as part of a file name, it cannot contain operating system special characters such as & ; / \ * ?
Primary MOVES DB	Name of the primary MOVES database to be used with the RunSpec. If blank, the created RunSpec will use the "Custom Input Database" (as set on the Advanced Performance Features panel) of the current RunSpec, and if that is blank the created RunSpec will use the MOVES default database configured at the time it is executed. This input will never change the name of the Default Database in the MOVES Configuration File as shown in the pull down menu item "Settings – Configure MOVES".
Output DB	Name of the database to store the created RunSpec's output. If blank, the created RunSpec will use the output database of the currently loaded RunSpec, and if that is blank an error message will be shown and processing stopped.
Domain DB	When using County or Project domains, MOVES requires an input database specific to the domain. If blank, the model will use the database named in the RunSpec. If neither is available, an error message will be shown and processing stopped.
Advanced Features DB	If the "Copy Saved Generator Data" checkbox is marked on the Advanced Performance Features panel, this field specifies the database where the data should be saved.
User DB	The MOVES GUI allows users to supply their own custom data using the Manage Input Data Sets panel. On that panel, zero or more user databases are listed. Such databases can be specified in the Control File in the User DB column and the columns to the right of it. For example, if two databases are desired, fill the User DB column with the first database, and fill the next column with the second.

MOVES places no limit on the number of such user databases.
---

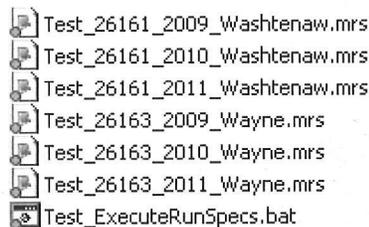
After successfully creating a template file, populating the file and saving it as a Control File, the user should return to the **Multiple RunSpec Creator** panel and browse for and select the Control File. The name of the Control File will be echoed in the Multiple RunSpec Creator panel to the left of the first Browse button.

The next step is for the user to enter a file name prefix in the File Name Prefix box. This input is a mandatory, 150 character limit, field that gives a prefix to the file name of each created RunSpec and .BAT file. Good usage of this field, perhaps with a project number or metropolitan area name, will help preserve the usefulness of the created RunSpecs.

After setting a prefix, use the second Browse button for an output directory to hold the created RunSpecs and .BAT file. Remember that many files may be created, so it may be best to create a new directory to hold the new files.

The Create RunSpecs button will check input values (control file name, file prefix, and output directory), read the control file, report errors, and if there are no errors, create the BAT and RunSpec files. Files are only generated if there are no errors.

As an example, using the above control file and a prefix of "Test", the files created are:



As can be seen, the naming convention is:

```
<prefix>_<countyID>_<year>[_<additional text>].mrs
<prefix>_ExecuteRunSpecs.bat
```

**Note** If RunSpecs are for the same county and year, it is important that the additional text be unique, otherwise, not all listed RunSpecs will be created.

The resulting BAT file from the example file is:

```
@echo off
rem Script generated by the MOVES Looping Tool
rem Based on control file:
rem C:\EPA\MOVES\MOVESGHGSource\Task910LoopingTool\ControlFile.xls
rem -----
rem Command Line Notes:
rem "-e DEBUG" logs all types of messages to MOVESbatch.log
rem -----
echo Changing to the MOVES folder and compiling code...
C:
cd "C:\EPA\MOVES\MOVESGHGSource"
call setenv.bat
```

```

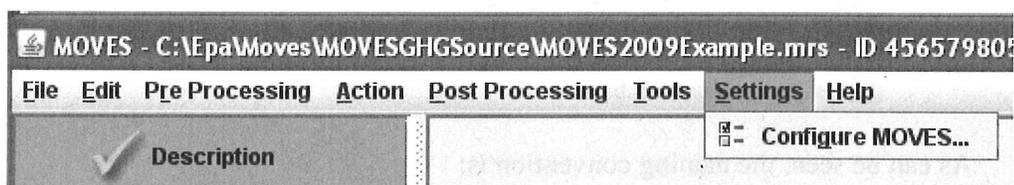
call ant compile
rem -----
echo Running Test_26161_2009_Washtenaw.mrs
java gov.epa.otaq.moves.master.commandline.MOVESCommandLine -e DEBUG -r
"C:\EPA\MOVES\MOVESGHGSource\Task910LoopingTool\Test_26161_2009_Washtenaw.mrs"
echo Running Test_26163_2009_Wayne.mrs
java gov.epa.otaq.moves.master.commandline.MOVESCommandLine -e DEBUG -r
"C:\EPA\MOVES\MOVESGHGSource\Task910LoopingTool\Test_26163_2009_Wayne.mrs"
echo Running Test_26161_2010_Washtenaw.mrs
java gov.epa.otaq.moves.master.commandline.MOVESCommandLine -e DEBUG -r
"C:\EPA\MOVES\MOVESGHGSource\Task910LoopingTool\Test_26161_2010_Washtenaw.mrs"
echo Running Test_26163_2010_Wayne.mrs
java gov.epa.otaq.moves.master.commandline.MOVESCommandLine -e DEBUG -r
"C:\EPA\MOVES\MOVESGHGSource\Task910LoopingTool\Test_26163_2010_Wayne.mrs"
echo Running Test_26161_2011_Washtenaw.mrs
java gov.epa.otaq.moves.master.commandline.MOVESCommandLine -e DEBUG -r
"C:\EPA\MOVES\MOVESGHGSource\Task910LoopingTool\Test_26161_2011_Washtenaw.mrs"
echo Running Test_26163_2011_Wayne.mrs
java gov.epa.otaq.moves.master.commandline.MOVESCommandLine -e DEBUG -r
"C:\EPA\MOVES\MOVESGHGSource\Task910LoopingTool\Test_26163_2011_Wayne.mrs"

```

This .BAT file can be run as is (via the command line or by double-clicking), or can be edited manually for distribution onto multiple MOVES master machines.

### 2.3.7 Settings

The Settings menu includes one option, **Configure MOVES**. This menu option allows the user to set the basic MOVES operating configuration.



#### 2.3.7.1 Configure MOVES

 A screenshot of the "Configure MOVES" dialog box. It has a title bar with "Configure MOVES" and a close button. The dialog is divided into three sections:
 

- Default Input Database:** Contains a "Server:" text box with "localhost" and a "Database:" dropdown menu with "MOVESDB20091216" selected. To the right are "OK" and "Cancel" buttons.
- Default Output Database:** Contains a "Server:" text box with "localhost".
- Shared Distributed Folder Path:** Contains a text box with "C:\Epa\Moves\MOVESGHGSource\SharedWork" and a "Browse..." button.

First, the user can specify the default input database by identifying the default server on which the MOVES input database is stored and the name of the default database. In most cases,

"localhost" is the desired server, although the user may input an alternate. Next the user can choose from a drop-down list to define the default database to be used as a data source for the MOVES RunSpec. In most cases, the user will choose the default database distributed with MOVES2010. The user also can select a server to house the default output database server under the **Settings**→**Configure** menu option; "localhost" is appropriate in most cases. The actual output database file is named on the **General Output** panel to specify output parameters as part of the MOVES RunSpec setup process.

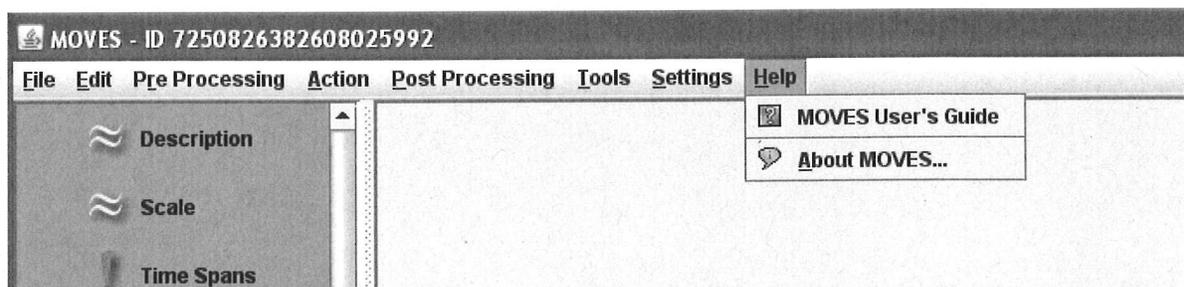
**Caution!** If the **Default Input Database** is changed the program should be stopped and restarted. This is because the user interface program will have already connected to the previously specified database in order to construct the GUI displays; thus the GUI may not match the new input database.

Using the **Settings** menu option, the user also can identify the **Shared Distributed Folder Path** for the MOVES Worker and MOVES Master program modules. This is the Windows folder where all intermediate and internal MOVES work files are stored during processing. This folder is installed as part of the MOVES program installation. The user should browse his system to find where this folder was installed and indicate the exact location. It may not be the same path as shown in the example above.

**Tip** By changing the **Shared Distributed Folder Path**, the user can take advantage of distributed processing by having the master and multiple worker computers on the same network "point" to a drive and folder that is accessible to all the computers involved.

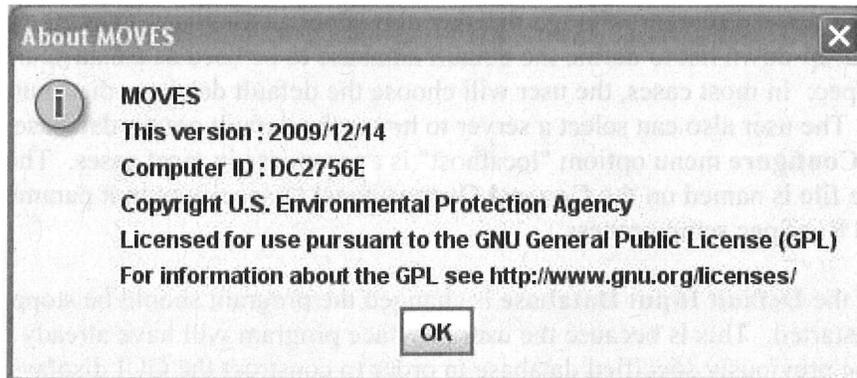
### 2.3.8 Help

Help provides access to .pdf versions of the MOVES User Guide and supplies general information about the version of MOVES being used.



Choose **MOVES User Guide** to open the document in .pdf format.

Choose **About MOVES** in the **Help** menu to obtain the release date of the model. This indicates which version of MOVES is open. In addition to the version, the EPA copyright and GNU General Public License (GPL) information is provided. The GNU license grants the user, free of charge, complete access to the object and source code of the MOVES program for personal use. For specific details regarding the GNU license, the user should consult the website <http://www.gnu.org/licenses/>.



## Appendix A – Using MOVES to Generate Lookup Tables

MOVES can calculate emission inventories (in total quantity of emissions for a given time) or emission rates (emissions divided by distance or population), depending on the user selection of “**Inventory**” or “**Emission Rates**” in the “**Calculation Type**” section of the “**Scale**” panel.

When modeling a single time and place, the emission rate calculations in MOVES are significantly more time consuming and require more computer memory than inventory calculations. However, for large scale projects, a user may want to use the emission rate calculations to create a lookup table of rates that--though a post-processor--can be applied to many times and places, thereby reducing total MOVES run time. Successful application of this approach requires a clear understanding of the rates calculations in MOVES and careful planning. EPA is developing tools to automate the use of this feature to create inputs for EPA’s Sparse Matrix Operator Kernel Emissions (SMOKE) model. This section describes the process more generally.

### Defining a Scenario

A scenario defines the set of conditions to which the emission rates may be applied. A scenario typically defines a specific fuel mix and set of Inspection/Maintenance (I/M) programs. It also assumes a certain pattern of trip activity (number of trips per vehicle per day and the distribution of those trips in time).

Some characteristics of the scenario depend on user choices: If a given age distribution is part of the scenario definition, that is, if the lookup table is to be applied only to areas with one particular age distribution, the lookup output can be smaller because the user does not need output by model year. On the other hand, detailing output by model year allows users to apply the lookup table to areas with diverse age distributions.

A scenario generally does not imply specific temperatures or speed distributions. Instead, these characteristics are key fields in the output used to match the emission rate with the appropriate vehicle activity.

### The Emission Rate Output Tables

Selecting “**Emission Rates**” on the **Scale** panel populates three additional output tables: **RatePerDistance**, **RatePerVehicle** and **RatePerProfile**. The **RatePerDistance** table stores emission rates for emissions that occur while a vehicle is in regular operation. The **RatePerVehicle** and **RatePerProfile** tables store emissions that occur while a vehicle is stationary, on “off network” roadtypes. The latter two tables differ because the emissions in the **RatePerVehicle** depend on the hour and the temperature while emissions in the **RatePerProfile** table also depend on temperatures in previous hours.

**1) RatePerDistance:** Includes emissions for the processes that occur while vehicles are operating: running exhaust (includes emissions during normal idle at traffic signals, etc), tire wear, brake wear, evaporative permeation, evaporative fuel vapor venting, evaporative fuel

leaks, crankcase running exhaust, refueling displacement and refueling spillage. The hour, day and month will not be relevant for most uses because the emissions are normalized with regard to activity. For national and county level runs, roadtype and avgSpeedbin will be relevant. For project-level runs, linkID will be important.

**Fields:**

MOVESScenarioID  
MOVESRunID  
yearID  
monthID  
dayID  
hourID  
linkID  
pollutantID  
processID  
sourceTypeID  
fueltype ID  
model year ID  
roadtype ID  
avgSpeedBinID  
temperature  
relhumidity  
RatePerDistance

**2)RatePerVehicle:** Includes emissions for most processes that occur while vehicles are stationary : start exhaust, start crankcase, permeation, liquid leaks, and extended idle (long haul combination trucks only). The month will not be relevant for most uses since, in rate calculations, it serves primarily as an identifier for the fuel mix and will be redundant with MOVESScenarioID, but daytypeid and hourid are important key fields since start and other activity vary per vehicle per hour.

**Fields:**

MOVESScenarioID  
MOVESRunID  
yearID  
monthID  
dayID  
hourID  
zoneID  
pollutantID  
processID  
sourceTypeID  
fueltype ID  
model year ID  
temperature  
RatePerVehicle

**3) RatePerProfile:** Includes emissions from the vapor venting process when vehicles are stationary. The daytypeid, hourid and temperatureprofileid are important key fields because the vapor venting emissions vary depending on activity and previous temperatures.

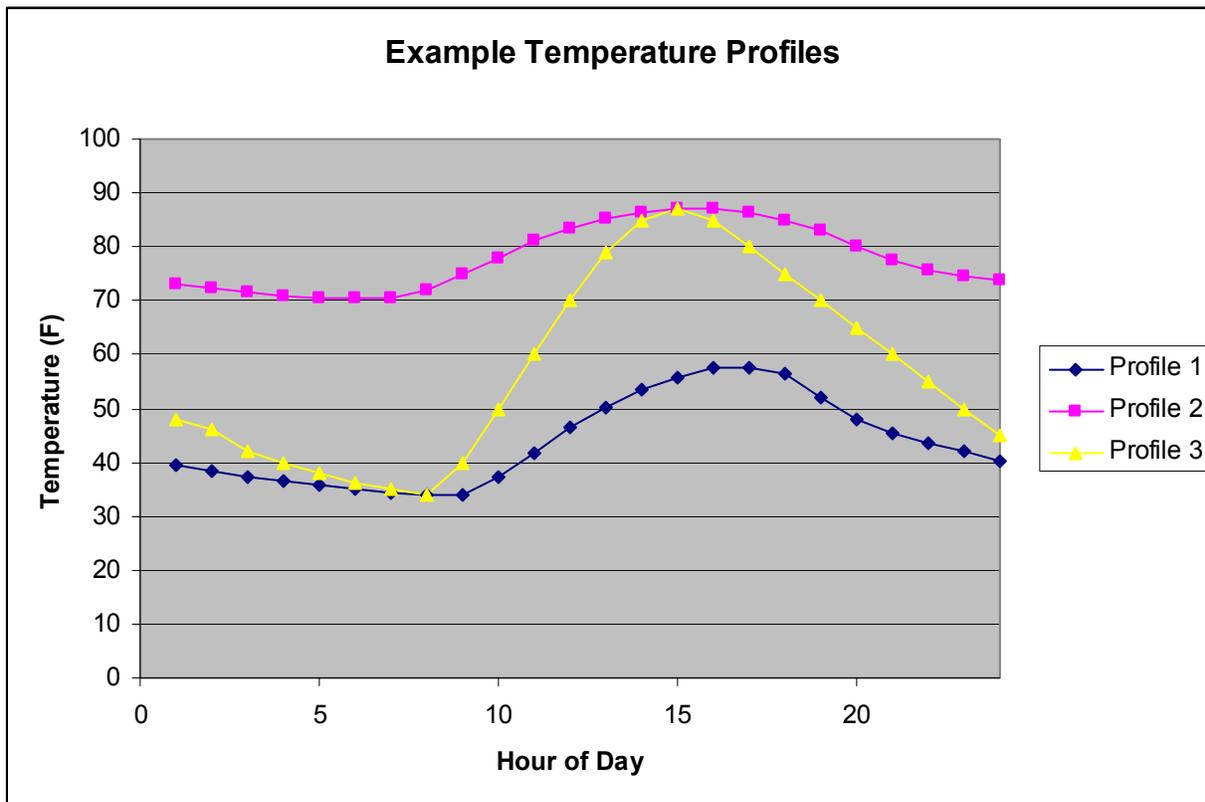
**Fields:**

MOVES Scenario ID  
MOVESRunID  
TemperatureProfileID  
yearID  
monthID  
dayID  
hourID  
pollutantID  
processID  
sourceTypeID  
fuelTypeID  
modelyearID  
temperature  
RatePerVehicle

### Setting Up Runs to Generate a Lookup Table

Setting up an emission rate lookup table for a single scenario requires a number of steps.

- 1) Define the scenario, and determine the counties, months, years and day-types in which it applies. Using the County Data Manager, a user can set up a run for the desired county, year, fuel supply, age distribution and I/M program.
- 2) Determine the temperatures and temperature profiles to be covered by the lookup table. A “temperature profile” is a set of 24 temperatures over the course of a day. If the user is modeling NOx, it may also make sense to include relative humidity as a variable.



3) Determine which pollutants and processes are of interest. Note that the modeling of hydrocarbons is more complicated than the modeling of other pollutants due to the need to include evaporative processes, and thus the need to represent temperature profiles as well as hourly temperatures.

4) Organize inputs to generate the most rates in the shortest total runtime, while maintaining a manageable output size. If multiple similar calculations are required, it is generally a good idea to combine them in the same run. However, due to the different requirements for different emission processes, redundant calculations can often be avoided by splitting the task into multiple runs.

For running exhaust, crankcase running exhaust, and refueling, hour doesn't matter, so 24 temperatures can be modeled as different hours in a single month. If more than 24 distinct temperatures are to be modeled, additional temperatures can be modeled using additional months (as long as the fuel characteristics are set to be the same for each month). If this approach is used, the month field in the output essentially becomes a label for a group of temperatures, and does not represent an actual calendar month.

For start, crankcase start exhaust, liquid leaks and evap permeation, hour and temperature matter. Sets of temperatures can be entered as individual "months". Again, this creates output where "month" represents a group of temperatures rather than a calendar month. In the illustration above, the three temperature profiles could be modeled in a single run as three nominal months, one for each profile. Note, however, that Profile 3 shares an hour/temperature point with the each of the other profiles (in hour 8 and hour 15). For a lookup table with greater scope, it could be efficient to design the runs to eliminate such redundancy.

Vapor venting emissions depend on hour, temperature and previous temperatures. For these emissions we must model each profile as a separate month. Thus, within a given MOVES scenario, monthid becomes equivalent to temperatureprofileid. In some cases, it may be appropriate to “bin” similar temperature profiles and to apply the results from one profile to a group of similar profiles.

Temperatures (and humidities) can be entered into the MOVES ZoneMonthHour table, using MySQL commands or through the County Data Manager. See User Guide section **2.3.3.4.1 Meteorology Data Importer** for more information on entering this data.

**⚠ CAUTION!** When producing Rates table for evaporative emissions, you must include at least one “running” emission process in the run specification because there are evaporative emissions that are reported only in the RatePerDistance table, but this table is populated only when a running emission process is selected. We plan to improve this in future versions of MOVES.

**!Tip** If “Emission Rates” are chosen on the **Scale** panel, output should be differentiated by “Source Use Type”. Doing so allows VMT, Road Type Distribution, and Average Speed Distribution to become placeholders (i.e., they must still be imported, but their values do not impact the results); however, if output is not differentiated by source type, the emission rate becomes a weighted average of the source types selected in the RunSpec and the three inputs mentioned will impact the results.

## Running MOVES to generate Lookup Tables

MOVES can generate lookup table output using the menu-driven interface, but for large-scale modeling, users will probably want to automate execution of a series of runs. See **Appendix C** for information on running MOVES from the command line. See **Section 2.3.6.1 Multiple Run Spec Creator** for information on a tool for creating a set of similar run specifications.

## Applying MOVES Emission Rate Output

To calculate an emissions inventory from emission rate output, rates in RatePerDistance values need to be multiplied by the appropriate Vehicle Miles Travelled (VMT), and rates in RatePerVehicle and RatePerProfile must be multiplied by appropriate vehicle population values. Doing this correctly can require a number of considerations.

- Rather than simply summing all activity in a speed or temperature bin, it is usually appropriate to do an interpolation as the rates are applied. For example, to calculate the total emissions from 100 miles of VMT with an average speed of 17.6 mph, a simplistic approach would be to multiply all of the VMT by the emission rate for speed bin 5 (17.5 mph-22.5 mph), however this approach is overly sensitive to even very small changes in speed. To reduce these boundary issues, users should instead interpolate between speed

bins (in this example, between the rates for speed bin 4 (12.5-17.5) and speed bin 5) based on the average speeds for those speed bins. .

- Depending on how the runs were organized, the user should take care not to double-count emissions with duplicate emission rates (for example, if the three profiles illustrated above were each modeled as independent months, there would be duplicate values in the ratepervehicle table for 8am at 34F, and for 3pm at 87F). A simple join of the rates and population table using hour and temperature would create extra rows that the user would need to be careful not to include in a sum.
- The user should be aware that evaporative emissions are stored in all three tables, and all three need to be used to fully account for evaporative emissions that occur during both driving and parking. The rates in RatePerDistance need to be multiplied by the appropriate VMT, and the rates in RatePerVehicle and RatePerProfile need to be multiplied by the appropriate population. They can then be summed to calculate the total emissions.

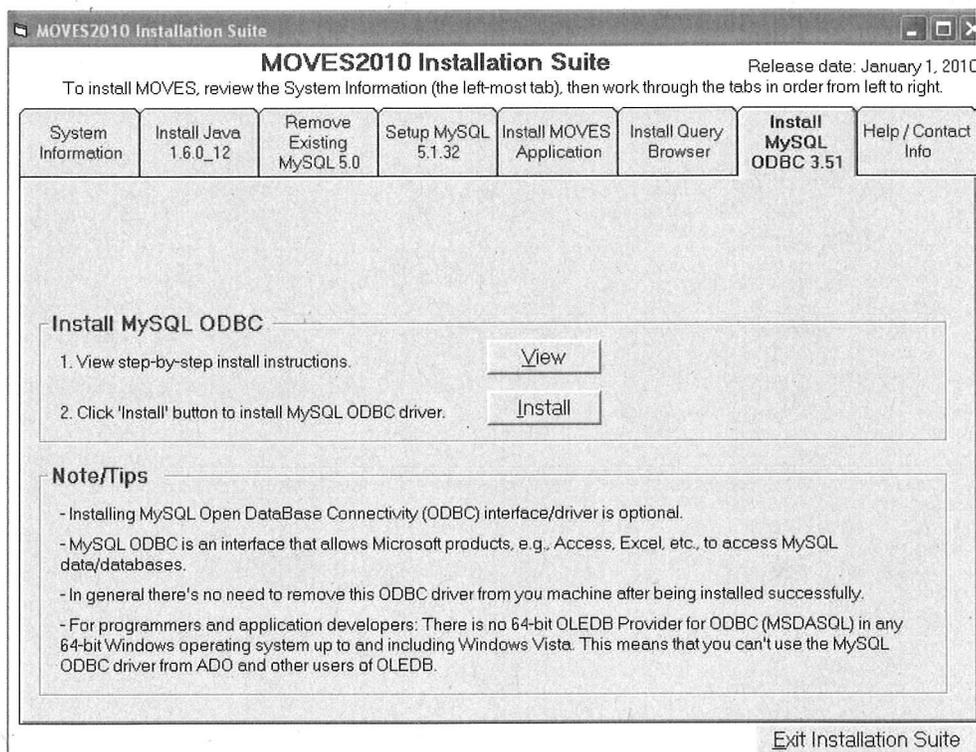
*[This page intentionally left blank.]*

## Appendix B - Linking MySQL Tables from MS ACCESS or MS Excel (includes ODBC usage)

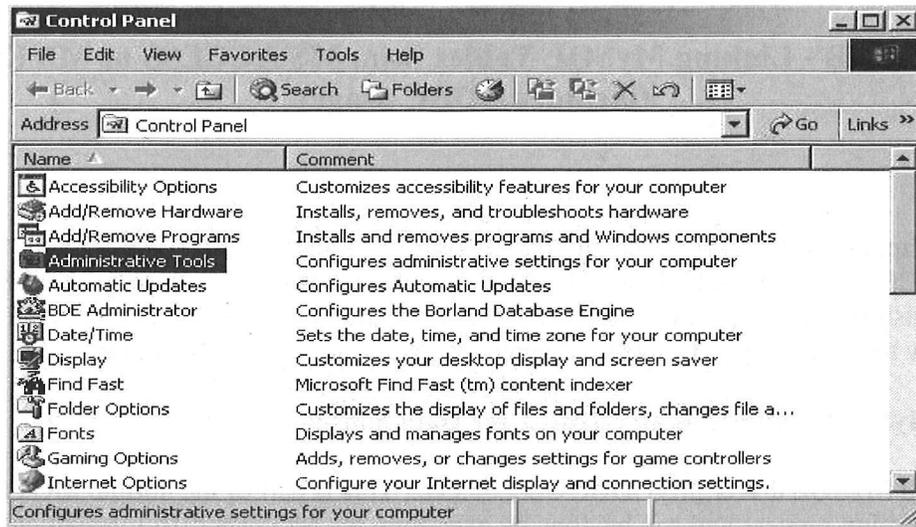
This appendix describes in detail how the user can transform MOVES MySQL tables into Microsoft ACCESS database tables or Microsoft Excel spreadsheets. This may be useful for those who wish to view or manipulate these tables in an alternate format. This appendix also explains how to use the Open Database Connectivity interface protocols and tools.

### Adding a MySQL database as an ODBC User Data Source

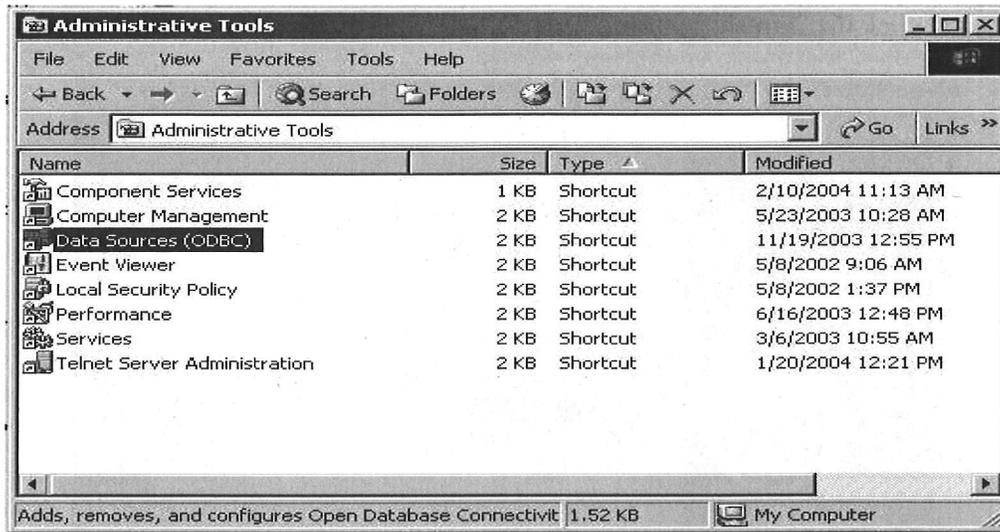
1. Install MySQL driver, MyODBC-3.51.06.exe. This is part of the normal MOVES installation process and can be done by executing the MOVES2010 Installation Suite



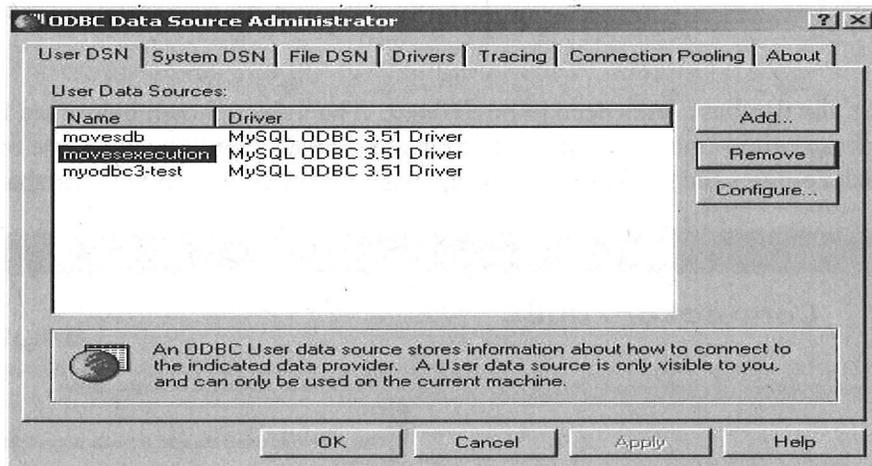
2. Create an ODBC data source with the same name as the folder which corresponds to the MySQL database.
  - a. In the Control Panel, select "Administrative Tools":



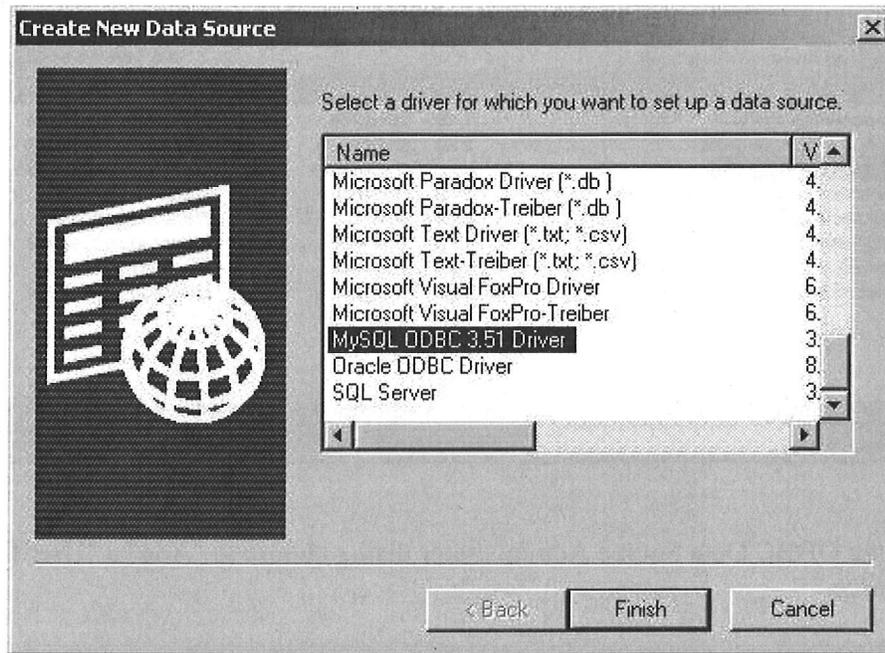
- b. In the "Administrative Tools" menu, select "Data Sources (ODBC)":



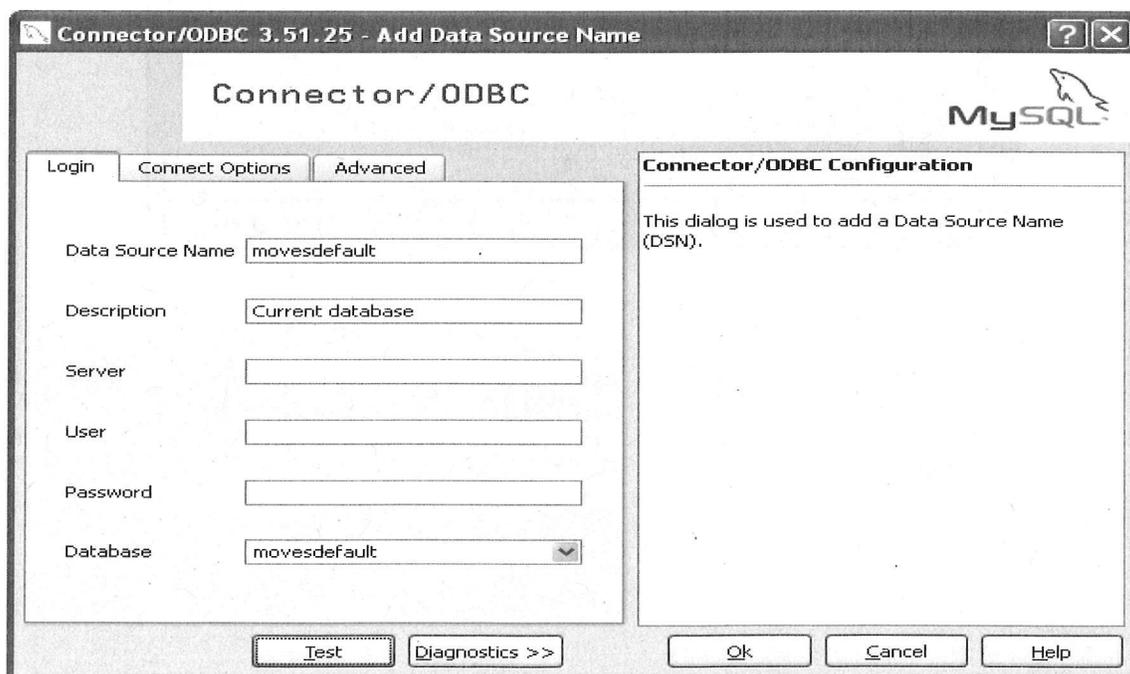
- c. In the ODBC Data Source Administrator dialog choose to "Add" a "User Data Source."



- d. The "New Data Source" should use the MySQL ODBC Driver that was installed earlier. Scroll through the list until the MySQL ODBC driver is found, highlight it, and click the "Finish" button.



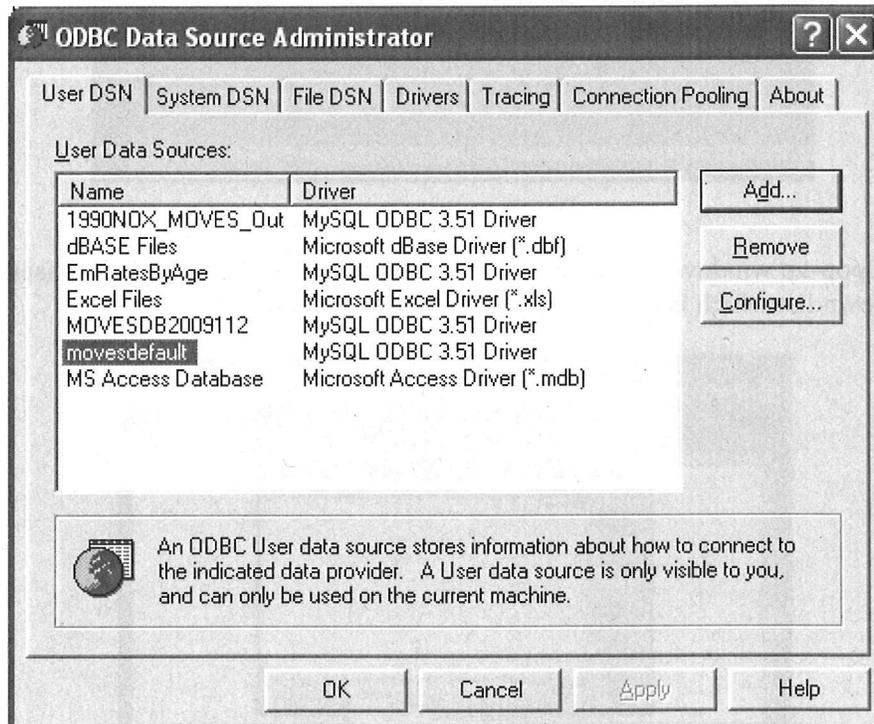
- e. Next, the database links need to be defined. The values shown below are for a database called "movesdefault". The database name should be the name of the folder which contains the desired MySQL database files (i.e., the MySQL database name).



- f. Once the database name and the data source name list an existing MySQL database name, the database connection can be checked by clicking on the "Test Data Source" button.

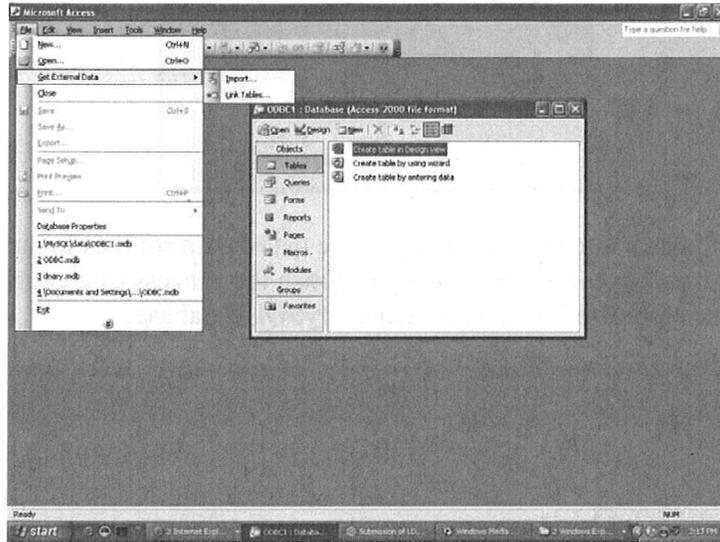


- g. The added data source will show up in the list of data sources in the ODBC Data Source Administrator dialog. This will finish the process of creating a new ODBC data source associated with a particular MySQL database.

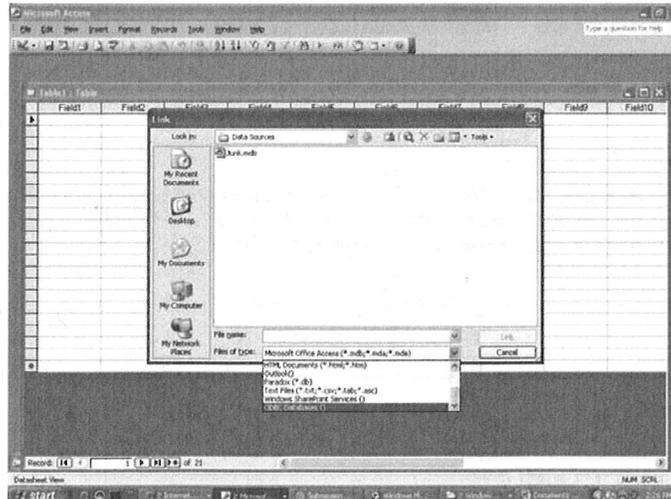


### Using Microsoft Access with MySQL Tables

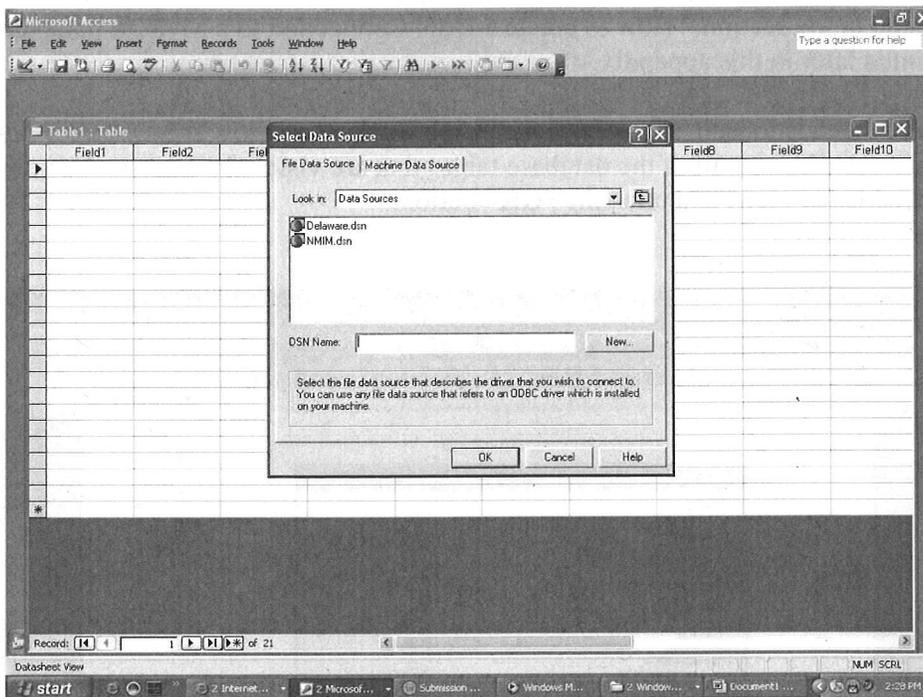
1. Run an Access .mdb (either a new one or an existing one).
2. On the top menu bar, look for menu items **"File"**, **"Get External Data"**, then **"Link Tables ..."** (see picture below). Click on the item **"Link Tables..."**



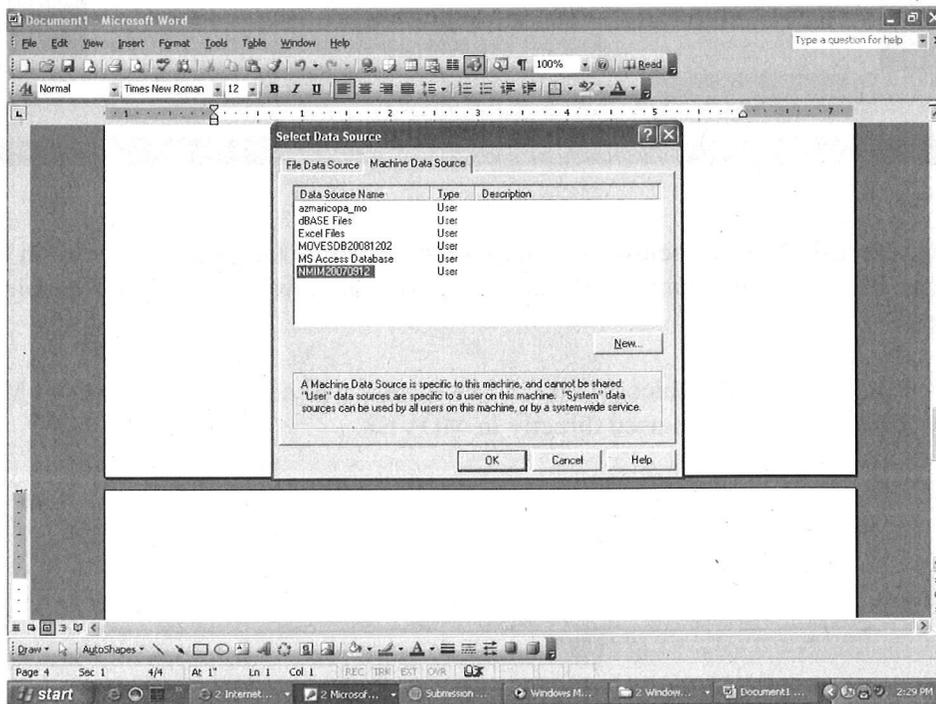
3. On the pop-up window like the one shown below, choose **"ODBC Databases()"** from the drop-down list which is located next to the label **"Files of type"**.



- Another pop-up window similar to the one below will appear. Click on the tab **Machine Data Source**, to select a database of interest.

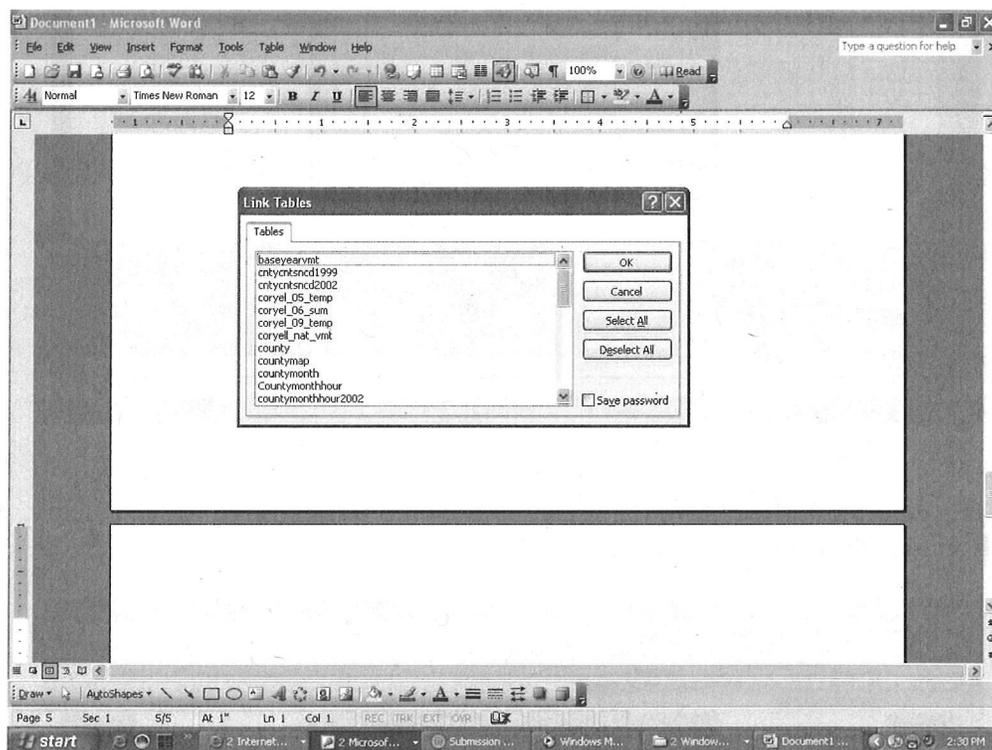


In this example NMIM20070812 is chosen from the **Data Source Name** column. Select **OK** to load the data into Access.



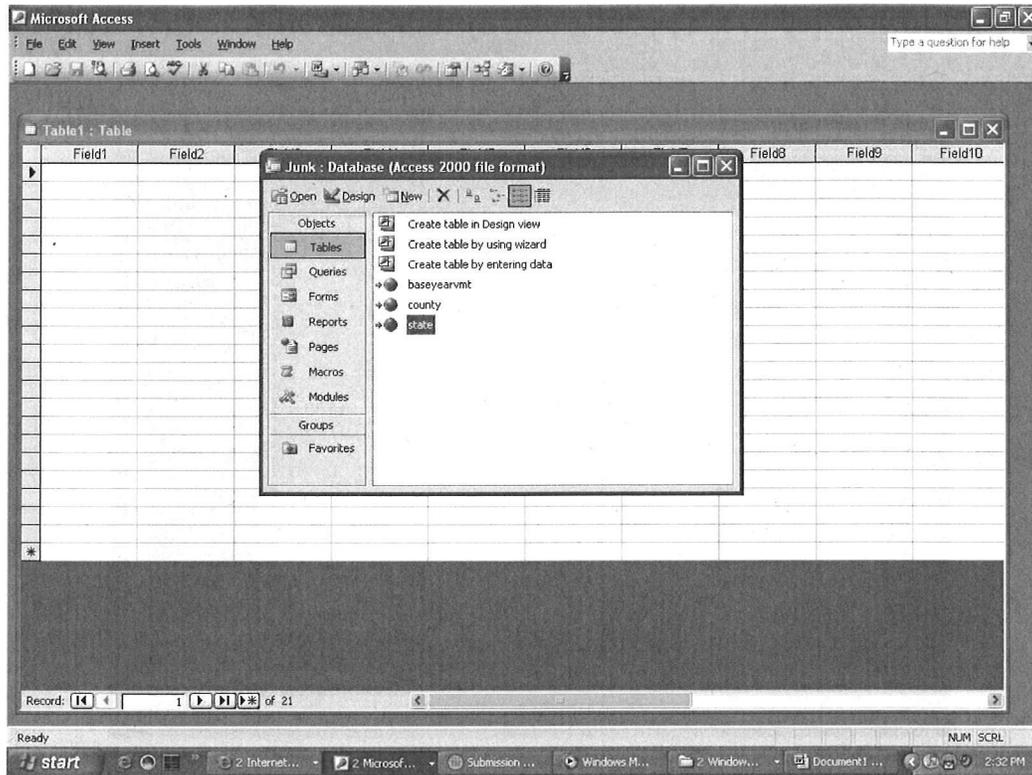
If the **Data Source Name** of the file is not listed, the user must follow the instructions provided in the section, "How to create a DSN Connection to a MySQL database" presented later in this appendix.

5. Once the data is loaded into access, the "**Link Tables**" window shown below will appear. This screen displays all of the database tables that are visible in Access. The user may select tables to view, and click the "**OK**" button.

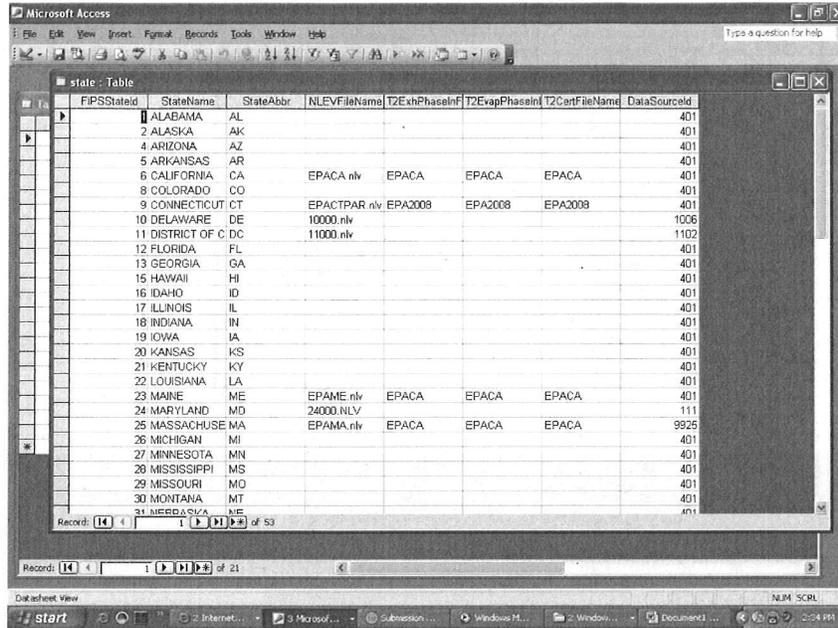


In the screenshot shown below, the user has chosen three tables to work with in MS Access: baseyearvmt, county, and state. The user can now work with these tables in MS Access.

**Caution!** Edits made to the tables or data in MS Access will not affect the original MySQL tables and cannot be used directly in MOVES.

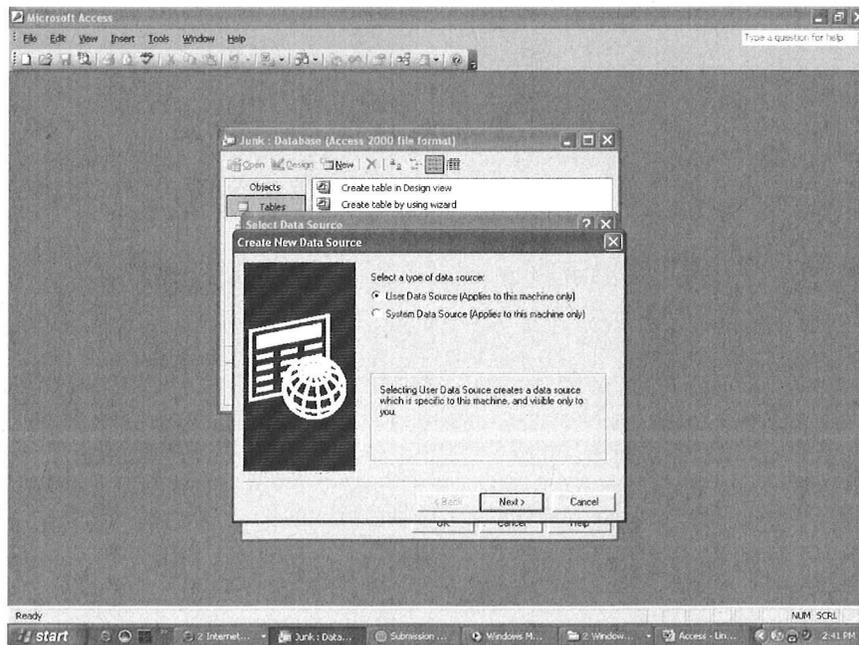


The user can view data specific to a particular state by clicking on the StateName in the table.

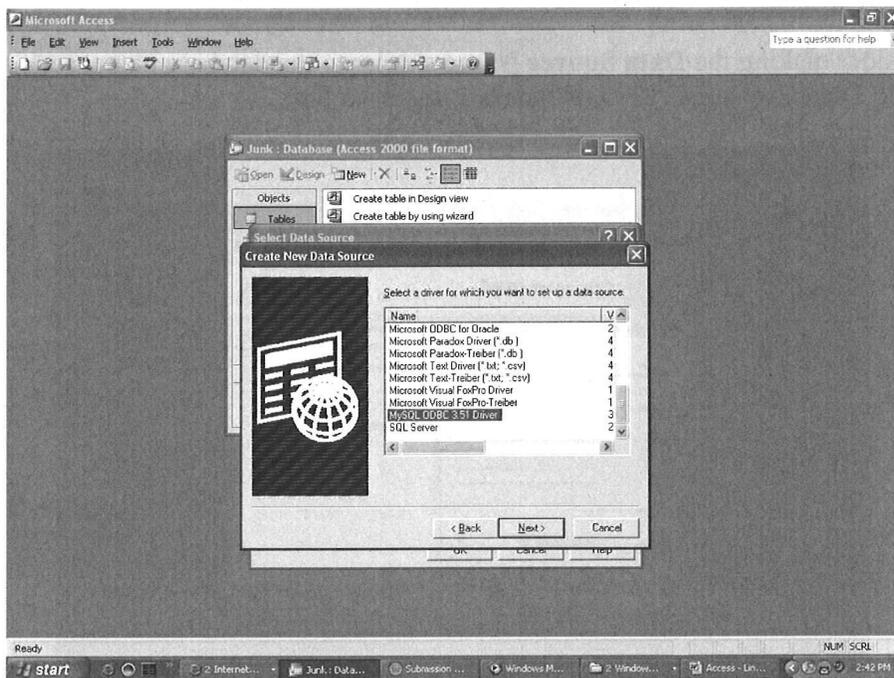


**How to create a DSN Connection to a MySQL database.**

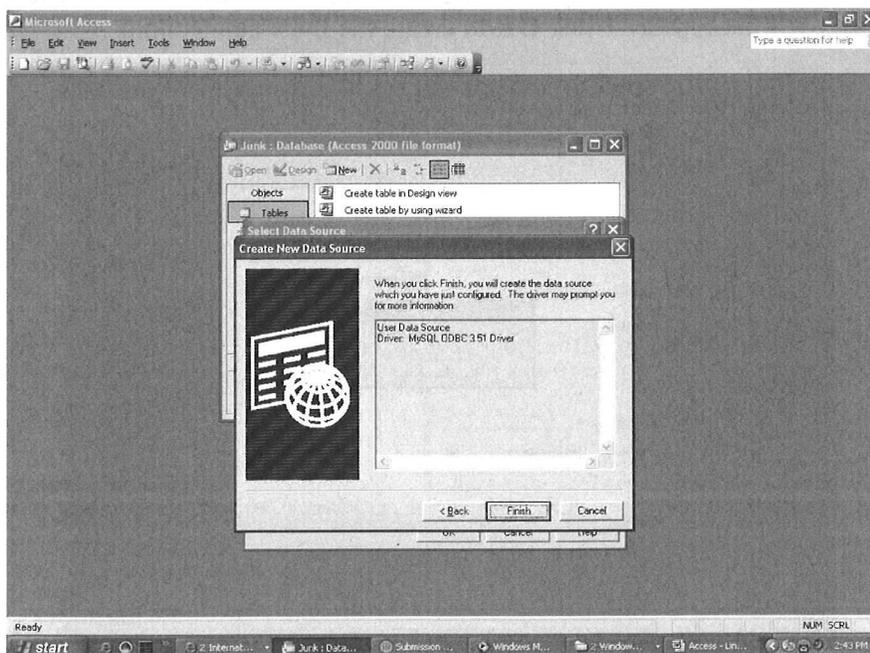
If the user does not find the MySql database referenced in the **Data Source Name** listing in MS Access, a DSN connection must be established. In order to do so, click on the **Machine Data Source** tab from the **Select Data Source** screen. Click the **“New”** button, and select **“Next>”**.



At the Create New Data Source Dialog box, select MySQL ODBC 3.51 Driver. Click “Next >”.

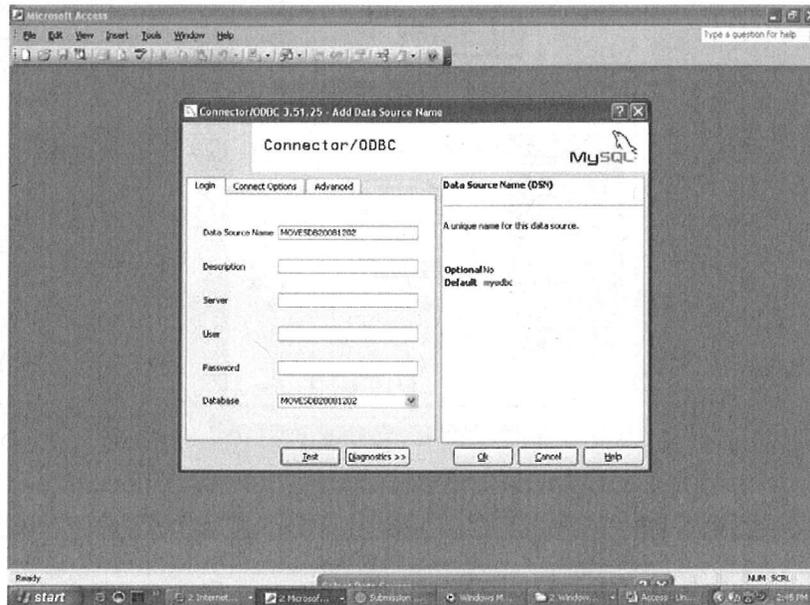


Click “Finish” to begin to establish the connection.

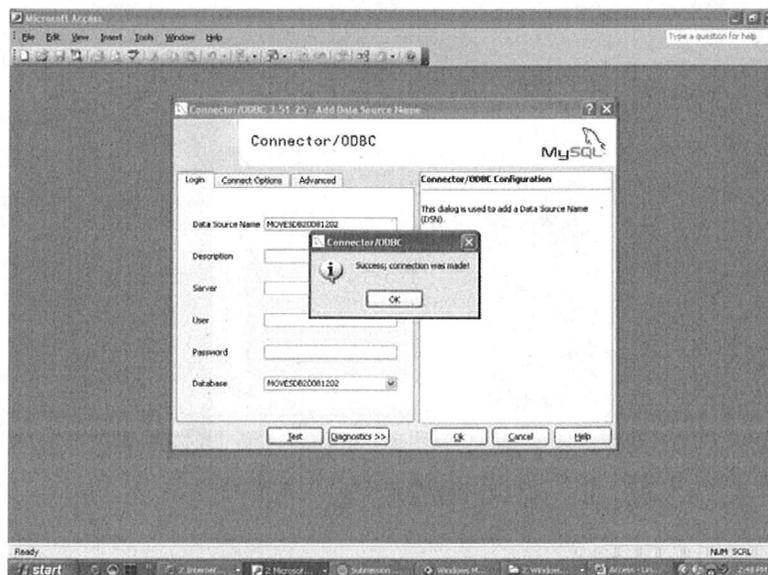


The **Connector/ODBC** screen is used to establish the connection from MS Access to the MySQL database. Use this screen to define the “**Data Source Name**”.

¶ **Note** Consider making the **Data Source Name** similar to the database name, such as **DSN\_DatabaseName**. This will make it easier to find.



Once the **Data Source Name** has been created, select the "**T**est" button. The popup box **Connector/ODBC** should open, with the message, "Success; connection was made!" Press **OK**, and the popup box will disappear. Click **OK** to create the DSN connection.

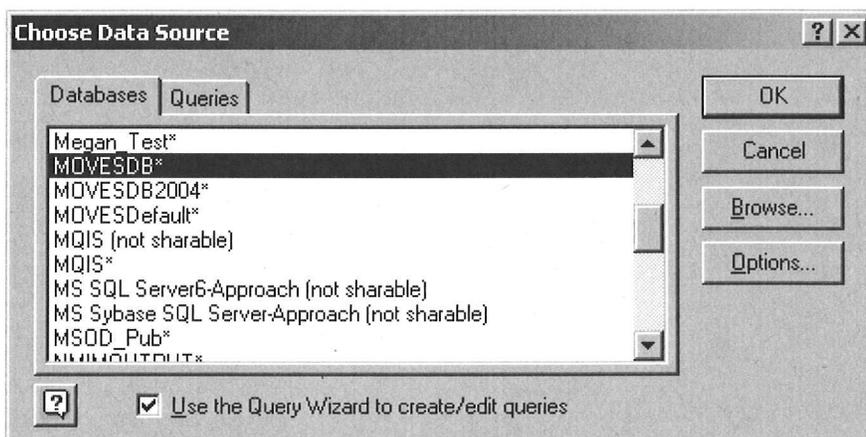


The **Data Source Name** will now be listed in the **Select Data Source** window.

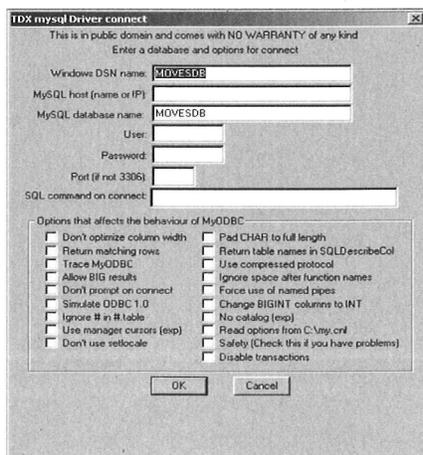
## Using Microsoft Excel with MySQL Tables

**Note** MySQL Query Browser can also be used to export MYSQL database tables to Excel.

1. Follow the instructions in the section "Adding a MySQL database as an ODBC User Data Source" for adding the desired MySQL database as an ODBC data source.
2. Open an Excel file. Click on "Data" from the top menu bar, and choose "**Import External Data**". Select "**New Database Query ...**" from the sub-menu window. This will open the "**Choose Data Source**" window.
3. From the **Choose Data Source** window, select the desired DSN from the list, e.g., "MOVESDB", then click "OK".



4. Click the "OK" button in the "**mysql Driver connect**" window.



5. Follow the instructions posted by the Wizard until finished.





## Appendix C - Using the Batch Command Line Interface

The MOVES command line allows MOVES to be executed without running its graphical user interface. It is useful in situations where repeated or unattended runs are needed, or when another computer program executes MOVES. This interface presumes that a MOVES run specification file has been prepared and that the user is running from the DOS prompt.

These instructions presume some familiarity with DOS commands. The DOS commands to execute the MOVES Command Line Interface are:

```
java MOVESCommandLine -r runspecfile
```

[OR]

```
java -XmxNNNm MOVESCommandLine -r runspecfile
```

[OR]

```
java MOVESCommandLine -rl runspeclistfile
```

[OR]

```
java XmxNNNm MOVESCommandLine -rl runspeclistfile
```

where

-XmxNNNm is an optional parameter and NNN is a three digit integer number. For example, -Xmx200m specifies you are asking java to allocate 200 megabytes of heap memory for your runs. This optional parameter is required only if your MOVES runs need more heap memory than the default determined automatically by java and your machine. The actual maximum amount of heap memory that you can ask for depends on your machine. You can increase or decrease the amount of heap memory as your machine allows. For example,

```
java -Xmx300m MOVESCommandLine -rl runspeclistfile
```

or

```
java -Xmx100m MOVESCommandLine -r runspecfile
```

"*runspecfile*" is the name of a file containing a saved MOVES run specification and "*runspeclistfile*" is the name of a text file containing a list of run specification file names, with one per line. The rest of the syntax is literal. The spelling of "MOVESCommandLine" is case sensitive because it is a java class name. When doing a batch run, you can identify the runspec list text file in the command line using quotes because it is located in a folder with spaces (e.g. "C:\My Documents"). However, the command line doesn't work if you put quotes in the runspec list file itself. The command line will think the file doesn't exist.

By executing one of these commands, the user runs the java interpreter (java.exe). MOVESCommandLine is a DOS parameter telling java.exe what java class file to begin executing, and the last two tokens are parameters passed to MOVESCommandLine.

Either version of this command can be executed from a DOS batch file, and batch files can contain multiple commands.

Prior to executing the command, the active directory should be set to the location where MOVES is installed, typically C:\Program Files\MOVES, and the SETENV.BAT file should be executed.

It can be difficult for DOS to find everything. For the command to work, three elements must be found:

- The java interpreter.
- The MOVESCommandLine java class.
- Any runspeclistfile and all runspecfiles.

Running SETENV.BAT insures that the java interpreter is found.

Running SETENV.BAT also insures that the procedure described in the next paragraph results in the MOVESCommandLine java class being found. Java experts can also use the CLASSPATH environment variable more directly to locate java class files.

If MOVES has been installed in the default location, C:\Program Files\MOVESyyyymmdd, then MOVESCommandLine.class is located at C:\Program Files\MOVESyyyymmdd\gov\epa\otaq\moves\master\commandline. This can be made the active directory, or the command can specify whatever part of the path is needed. For example, if the active directory is C:\Program Files\MOVESyyyymmdd, the command line interface class would be specified as "gov.epa.otaq.moves.master.commandline.MOVESCommandLine".

One way to insure that the runspecfile or the runspeclistfile is found is to specify the full path. If a simple file name is used the file should be located in the active DOS directory.

If the GUI is used to create an importer XML file (see Section 2.3.3.3), the XML file can be edited and executed via the command line. Remember that when using the command line, text output is not sent to the screen but instead stored in MOVESBatch.log. An example command line is:

```
java gov.epa.otaq.moves.master.commandline.MOVESCommandLine  
-i importAllFromTemplates.xml
```

Be sure to place all of this on one line. The "-i" option directs MOVES to the XML file that describes the import actions to occur. Note that this XML file contains a summary of the RunSpec within it, thus allowing wildcards even when importing via the command line. For MOVES installations on Windows 2000 platforms, there may be a problem with the DOS command string length not allowing more than 126 characters. To resolve this, move MOVES to

the root of C: drive (C:\MOVES\), so that the length of the command line string will be shorter. Note that Windows2000 is not a recommended nor supported environment in which to run MOVES.



## Appendix D - Creating an On-Road Retrofit Input File

The "On-Road Retrofit" strategy, described in Section 2.9.9 allows the user to enter information about diesel trucks and buses that have been retrofit with emission control equipment. An On-Road Retrofit Parameters File must be input to the MOVES model to model on-road vehicle retrofit effects. This file can be in tab-delimited text, comma-delimited (\*.csv), or Microsoft Excel (\*.XLS) format. The user must create the Retrofit Parameters File by using either a text editor such as Wordpad or Notepad, or a spreadsheet such as Excel.

### A. Header and Comment Rows

The first row of the Retrofit Parameter File must be a header row that contains the retrofit input parameter names separated by tabs. The remaining rows of the file contain the retrofit parameters. Comment rows are allowed if the first character in the row or the first cell in the tabbed file or \*.XLS file is the '#' character.

### B. Retrofit Parameter File Format

The Retrofit Parameter File must contain the following ten parameters in the order listed. Each parameter's possible values are provided.

#### 1. Retrofit Pollutant

This column must contain the full MOVES pollutant name or the pollutant abbreviation for which the user wishes to model retrofit data. (In MOVES2010 the user may not use the MOVES PollutantID number.) The data must appear exactly as shown in Table D-1 below, except that the pollutant abbreviation and/or full name input are not case sensitive.

**Table D-1**  
**Retrofit Parameter File Pollutants**

PollutantID	Full MOVES Pollutant Name	Pollutant Abbreviation
1	Total Gaseous Hydrocarbons	HC
2	Carbon Monoxide (CO)	CO
3	Oxides of Nitrogen	NO <sub>x</sub>
5	Methane (CH <sub>4</sub> )	CH <sub>4</sub>
6	Nitrous Oxide (N <sub>2</sub> O)	N <sub>2</sub> O
90	Atmospheric CO <sub>2</sub>	AT CO <sub>2</sub>
91	Total Energy Consumption	TotEnergy
92	Petroleum Energy Consumption	PetEnergy
93	Fossil Fuel Energy Consumption	FossilEnergy

(cont.)

**Table D-1  
Retrofit Parameter File Pollutants (cont.)**

<b>PollutantID</b>	<b>Full MOVES Pollutant Name</b>	<b>Pollutant Abbreviation</b>
98	CO <sub>2</sub> Equivalent	CO2EQ
101	Primary PM10 - Organic Carbon	PM10OC
102	Primary PM10 - Elemental Carbon	PM10EC
105	Primary PM10 - Sulfate Particulate	PM10Sulfate
106	Primary PM10 - Brakewear Particulate	PM10Brake
107	Primary PM10 - Tirewear Particulate	PM10Tire
111	Primary PM2.5 - Organic Carbon	PM25OC
112	Primary PM2.5 - Elemental Carbon	PM25EC
115	Primary PM2.5 - Sulfate Particulate	PM25Sulfate
116	Primary PM2.5 - Brakewear Particulate	PM25Brake
117	Primary PM2.5 - Tirewear Particulate	PM25Tire

## 2. Retrofit Emission Process Input

This column must contain the full MOVES process name or the ProcessID abbreviation for which the user wishes to model retrofit data. In MOVES2010, the user should not use MOVES ProcessID. The data must appear exactly as shown in Table D-2 below, except that the ProcessID abbreviation and full name are not case sensitive.

**Table D-2  
Retrofit Parameter File Processes**

<b>ProcessID</b>	<b>Full MOVES Process Name</b>	<b>Process Abbreviation</b>
1	Running Exhaust	Running
2	Start Exhaust	Start
90	Extended Idle Exhaust	Extended Idle
9	Brakewear	Brake
10	Tirewear	Tire

## 3. Retrofit Fuel Type Input

This column must contain the full MOVES fuel name or the fuel abbreviation for which the user wishes to model retrofit data. In MOVES2010, the user should not use FuelTypeID. The data must appear exactly as shown in Table D-3 below, except that the fuel abbreviation and full name are not case sensitive.

**Table D-3**  
**Retrofit Parameter File Fuel Types**

<b>FuelTypeID</b>	<b>Full MOVES Fuel Type Name</b>	<b>Fuel Type Abbreviation</b>
1	Gasoline	Gas
2	Diesel Fuel	Diesel
5	Ethanol (E85)	Ethanol

#### 4. Retrofit Vehicle Source Type Input

This column must contain the Sourcetype abbreviation or the full MOVES Sourcetype name for which the user wishes to model retrofit data. In MOVES2010, the user should not use SourceTypeID number. The data must appear exactly as shown in Table D-4 below, except that the sourcetype abbreviation and full name input are not case sensitive.

In MOVES2010, only diesel (FuelTypeID = 2) vehicles with the following SourceTypeIDs are covered by the Retrofit Calculations: 32, 41, 42, 43, 51, 52, 53, 61, and 62.

**Table D-4**  
**Retrofit Parameter File Source Types**

<b>SourceTypeID</b>	<b>Full MOVES Vehicle SourceType Name</b>	<b>SourceType Abbreviation</b>
11	Motorcycle*	MC
21	Passenger Car*	Car
31	Passenger Truck*	PTruck
32	Light Commercial Truck	ComTruck
51	Refuse Truck	RefuseTruck
52	Single Unit Short-Haul Truck	SUShortTruck
53	Single Unit Long-Haul Truck	SULongTruck
54	Motor Home	MH
43	School Bus	SBus
42	Transit Bus	TBus
41	Intercity Bus	IBus
61	Combination Short-haul Truck	CShortTruck
62	Combination Long-haul Truck	CLongTruck

\* These sourcetypes should not be used in retrofit input files.

MOVES compares the user inputs for pollutant, process, fuel type, and source type with the stored retrofit possibilities and prompts the user if an incorrect name, abbreviation or number for pollutant, process, fuel type, or source type is entered.

### **5. Initial Calendar Year of Retrofit Implementation**

The Initial Calendar Year of the Retrofit Implementation is the first calendar year that a retrofit program is administered. Initial Calendar Year input must be equal to or less than the Final Calendar Year of Retrofit Implementation. All months within a calendar year are affected equally by the retrofit.

### **6. Final Calendar Year of Retrofit Implementation**

The Final Calendar Year of the Retrofit Implementation is the last calendar year that a retrofit program is administered. Final Calendar Year input must be equal to or greater than the Initial Calendar Year of Retrofit Implementation.

### **7. Initial Model Year that will be Retrofit**

The Initial Model Year that will be Retrofit is the first model year of coverage for a particular vehicle class/pollutant combination. Valid entries for initial model year must meet the following mathematical requirement:

$$\text{Initial Model Year} \geq \text{Initial Calendar Year} - 30$$

The Initial Model Year cannot be greater than the Final Model Year that will be Retrofit.

### **8. Final Model Year that will be Retrofit**

The Final Model Year that will be retrofit is the last model year of coverage for a particular vehicle class/pollutant combination. No retrofit will be performed on Final Model Year input which is larger than the Evaluation Calendar Year. Also, the Final Model Year input cannot be less than the Initial Model Year that will be Retrofit.

### **9. Percentage of the Fleet Retrofit per Year**

The Percentage of the Fleet Retrofit per Year represents the percentage of VMT of a particular fleet of a particular vehicle class, retrofit calendar year group, model year group, and pollutant combination that is to be rebuilt in a given calendar year. For a successful retrofit simulation, a value greater than zero and less than or equal to 100.0% must be entered. MOVES checks to ensure that the product of the number of calendar years of retrofit coverage (Final Calendar Year of Retrofit Implementation - Initial Calendar Year of Retrofit Implementation) and the Percentage of the Fleet Retrofit per Year does not exceed 100%. For example, a retrofit simulation will be flagged as invalid, and an error message will appear, if the simulation has a retrofit program start in calendar year 2005, a program end in calendar year 2008, and a yearly Fleet Retrofit Percentage of 50% ( $3 * 50\% > 100\%$ ).

## 10. Percentage Effectiveness of the Retrofit

The Percentage Effectiveness of the Retrofit is the percent emission reduction achieved by a retrofit. It is computed from a non-retrofit emission baseline. The user must enter a retrofit effectiveness value for a particular vehicle class, retrofit calendar year group, model year group, and pollutant combination. All values up to 100% are valid. A negative value is permitted because it implies an emission increase as a result of retrofit, which can occur. A value greater than 100% is not permitted because it implies negative emissions will be generated. See also EPA's guidance on quantifying benefits from retrofits at <http://www.epa.gov/otaq/stateresources/transconf/policy.htm>



## Appendix E - Best Practices

There are several “best practices” that can assist the user in managing various aspects of the MOVES process. In particular, we have found it helpful to be careful and consistent in naming, managing & documenting RunSpecs and files.

**NOTE** MOVES database names must be consistent with MYSQL naming conventions. In general, this means they may contain letters, numbers, "\_"(underscore) and "\$" characters. They may not contain spaces, special characters, or be "reserved words" (SQL words such as "select", "where", "delete", etc.).

**Table E-1  
MOVES Best Practices**

<b>Suggestion</b>	<b>Rationale</b>	<b>Example</b>
Name all your MOVES output databases with a code indicating that the database includes MOVES output	Prevents overwriting input files; makes it easier to identify output files.	Washtenaw2008_MO (for MOVES Output) or Washtenaw2008_out
When testing MOVES, name output with "Temp" or other indicator.	Mysql/data folder can get cluttered. This allows you to clean up more easily	Washtenaw2008_temp_MO
Name all your MOVES input databases and strategy files with a code indicating that the database contains MOVES alternate input data.	Helps you organize your inputs and prevents you from inadvertently damaging important input information.	Washtenaw2008_IN
Name all your MOVES run specification text files with a code indicating that the file contains a MOVES run spec.	Helps you organize your input information and can be used to locate files using a search.	Washtenaw2008.mrs (for MOVES Run Specification) or Washtenaw2008_mrs.txt
Name all the files you use for a specific run using the same name, but with extensions to identify the parts (see file naming suggestions above).	Helps you organize your information.	Washtenaw2008_in, Washtenaw2008.mrs, Washtenaw2008_mo