# Change Log for NCD20140620\_nei2011v2

June 2014

NCD20140620\_nei2011v2, created by ERG, is a copy of the NCD20130531\_nei2011v1 database with modifications made to incorporate new and updated external data files provided by the Delaware Department of Natural Resources and Environmental Control.

The following changes were made:

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Notes** | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external files. Updated entries were given a datasourceid = 9968 | |
| **Datasource** | Inserted datasourceid 9968 for Delaware Department of Natural Resources and Environmental Control | |
| **External Files** | 10000.POP | Updated |
| 10000AIR.ALO | Updated |
| 10000CEN.ALO | Added |
| 10000COM.ALO | Added |
| 10000CON.ALO | Added |
| 10000FRM.ALO | Added |
| 10000GC.ALO | Updated |
| 10000HOU.ALO | Updated |
| 10000LOG.ALO | Updated |
| 10000LSC.ALO | Added |
| 10000MFG.ALO | Added |
| 10000MIN.ALO | Added |
| 10000OIL.ALO | Added |
| 10000RR.ALO | Added |
| 10000RVP.ALO | Updated |
| 10000SBC.ALO | Added |
| 10000SBR.ALO | Added |
| 10000SNM.ALO | Added |

# Change Log for NCD20130531\_nei2011v1

May 2013

NCD20130531\_nei2011v1, created by ERG, is the database used for the nei2011v1.

The following steps describe how NCD20130531\_nei2011v1 was built.

1. This database is a copy of NCD20130531 with the following modifications:
   1. Replace the tables countyyearmonth, gasoline, diesel, and countymonthhour with those supplied by EPA for 2011. The files provided by EPA were: RegionalFuels\_2011\_20130208fuelsNMIM.zip and countymonthhour2011.zip. (This means state-supplied fuel and met data were not used in the nei2011dv1 in favor of EPA -supplied data.)
   2. The 2011 fuel tables supplied by EPA did not contain fuel data for Alaska, Hawaii, Puerto Rico, or the U.S. Virgin Islands. Data from the countyyearmonth, diesel, and gasoline tables were updated using fuel data for all years for these four states from the corresponding tables in the NCD20121201a.

# Change Log for NCD20130531

May 2013

NCD20130531, created by ERG, is a complete record of data submitted for the 2011 NEI by states via the EIS Gateway as of May 31, 2013.

In addition, ERG worked with Davidson County, Tennessee, Texas, and New York to update some data inputs based on model inputs used to create the emissions submission for the 2011 NEI. The following is a brief description of these inputs:

1. Nashville/Davidson County Tennessee
   1. Fuel data from Nashville’s NONROAD option files were extracted and used to update RVP and sulfur values in the fuel data tables within NMIM.
2. New York
   1. Provided new allocation file for housing developments
   2. Fuel data from New York’s NONROAD option files were extracted and used to update RVP and sulfur values in the fuel data tables within NMIM.
3. Texas
   1. Updated external population file using population data extracted from the Texas Nonroad (TexN) Model
   2. Year-by-year population data from TexN were used to create a new growth file for Texas.
   3. Activity data from TexN were processed using a statistical analysis software program (SAS©). A weighted average activity value was calculated for each equipment SCC using horsepower-hours as the weighting factor. (HP-hours were selected as the weighting factor as this value should correlate reasonably closely with total exhaust emissions.) The first step in this process was to calculate the cumulative hp-hrs over the entire population. Next, the population and hp-hrs were summed over each unique SCC-DCE Subsector-County Flag-Load Factor combination. Then, the fraction of hp-hrs for each SCC within each DCE Subsector and County Flag was calculated and applied to the total activity value. The resulting SAS outputs were then formatted according to the external file format for activity used by NMIM.
   4. Fuel properties within TexN were used to update the fuel data within NMIM for gasoline RVP, diesel sulfur, gasoline sulfur, marine diesel sulfur, CNG and LPG sulfur, MTBE volume, ETBE volume, TAME volume, EtOH volume, and the market share for each of these volumes

This database is a copy of NCD20130331 with the following modifications:

* 1. Update the database using state-submitted data (submitted via EIS as of May 31, 2013).
  2. Update the database using state-originated data used to develop EIS emissions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Maryland** | | | |
| **Table Name** | **Notes** | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD file. Updated entries were given a datasourceid = 9964 | | |
| **Datasource** | Inserted datasourceid 9964 for Maryland Department of Environment | | |
| **External Files** | 24000.POP | Updated to reflect state-specific equipment population for pleasure craft. MDE collected Maryland’s pleasure craft data via National Marine Manufacturers Association (NMMA) web (http://www.nmma.org/assets/cabinets/Cabinet233/MD\_1and2.pdf) | |
| **New Jersey** | | | |
| **Table Name** | **Notes** | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD files. Updated entries were given a datasourceid = 9965 | | |
| **Datasource** | Inserted datasourceid 9965 for New Jersey DEP | | |
| **Gasoline** | Two gasoline specifications were added to the gasoline table and assigned numbers 340001 and 340002. These new IDs reflect:   * Updated EtOH volume * Updated Sulfur * Updated RVP * Averaged E200 & E300 from NMIM defaults for IDs 15 & 16 | | |
| **Countyyearmonth** | Revised to use the new gasoline specifications for datasource id 9965. Set RMDieselID to 236. Set NRDieselID to 32. Updated to use the two new gasoline IDs. | | |
| **Diesel** | Diesel fuel sulfur values and ID numbers 32 and 236 were added. | | |
| **External Files** | 34000.pop | | Updated to reflect state-specific equipment population for recreational marine vessel categories based on National Marine Manufacturers Association (NMMA) |
| **Davidson County (Tennessee)** | | | |
| **Table Name** | **Notes** | | |
| **Datasource** | Inserted datasourceid 9963 for Nashville Pollution Control Division | | |
| **Countyyearmonth** | Revised to use the new gasoline specifications for datasource id 9963 based on data harvested from the NONROAD option files submitted by the agency. Set RMDieselID to 236. Set NRDieselID to 32. Updated to use the four new gasoline IDs. New NRGasolineIDs were based on fuel properties by month:   * January (1) and December (12) set to NRGasolineID 4703704 * February (2), March (3), April (4), October (10), and November (11) set to NRGasolineID 4703703 * May (5) set to NRGasolineID 4703702   June (6), July (7), August (8), and September (9) set to NRGasolineID 4703701 | | |
| **Diesel** | Diesel fuel sulfur values and ID numbers 32 and 236 were added. | | |
| **Gasoline** | Four gasoline specifications were added to the gasoline table. These updated reflect new RVP and sulfur values based on the agency’s NONROAD option files used for the emissions estimations. | | |
| **New York** | | | |
| **Table Name** | **Notes** | | |
| **Datasource** | Inserted datasourceid 9966 for New York Department of Environmental Conservation | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD files. Updated entries were given a datasourceid = 9966 | | |
| **External Files** | 36000.pop | | Updated to reflect state-specific housing allocation |
| **Countyyearmonth** | Revised to use the new gasoline specifications for datasource id 9966 based on data harvested from the NONROAD option files submitted by the agency. Set RMDieselID to 1389. Set NRDieselID to 1218. Updated to use the new gasoline IDs (3600001 through 3600035). | | |
| **Diesel** | Diesel fuel sulfur value and ID number 1389 was added. | | |
| **Gasoline** | Thirty-five gasoline specifications were added to the gasoline table. These updated reflect new RVP and sulfur values based on the agency’s NONROAD option files used for the emissions estimations. | | |
| **Texas** | | | |
| **Table Name** | **Notes** | | |
| **Datasource** | Inserted datasourceid 9967 for the Texas Commission on Environmental Quality | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD files. Updated entries were given a datasourceid = 9967 | | |
| **External Files** | 48000.pop | | Population data from TexN for the year 2011 were summed by SCC and HP Bin and an average of “AvgHP” by SCC and HP Bin used to create a new population file to be read by NMIM. |
|  | 48000.grw | | Population data from TexN were summed by year and SCC. The growth was scaled using 2011 as the base year. Each SCC was assigned the appropriate indicator code according the default NONROAD mapping pattern for indicator codes. |
|  | 48000.dat | | Using data harvested from TexN, calculated the weighted average activity using hp-hrs as the weighting factor. |
|  | 48000.air | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.cen | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.com | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.con | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.frm | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.gc | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.hou | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.log | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.lsc | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.mfg | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.min | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.oil | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.rr | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.rvp | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.sbc | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.sbr | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.snm | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.wib | | Reflects allocation based on TexN population values for 2011 |
|  | 48000.wob | | Reflects allocation based on TexN population values for 2011 |
| **Countyyearmonth** | Revised to use the new gasoline specifications for datasource id 9967 based on data harvested from TexN. Set RMDieselID to 234. Set NRDieselID to 3, 4, 5, 6, 7, 8, or 15 according to county fuel properties. Updated to use the new gasoline IDs (4800001 through 4800094). | | |
| **Diesel** | Diesel fuel sulfur values and ID numbers 3, 4, 5, 6, 7, 8, and 15 were added. | | |
| **Gasoline** | Ninety-four gasoline specifications were added to the gasoline table. These updated reflect fuel data within the Texas Nonroad (TexN) model by county. These fuel properties within TexN were determined using state fuel sampling surveys conducted by TCEQ. The new gasoline fuel properties are assigned ID number from 4800001 through 4800094 and include updates to:   * Average market shares for MTBE, ETBE, TAME, and EtOH * Percent Volumes for MTBE, ETBE, TAME, and EtOH * RVP * Sulfur | | |

# Change Log for NCD20130331\_nei2011dv1

March 2013

NCD20130331\_nei2011dv1, created by ERG, is the database used for the nei2011dv1.

The following steps describe how NCD20130331\_nei2011dv1 was built.

1. This database is a copy of NCD20130331 with the following modifications:
   1. Replace the tables countyyearmonth, gasoline, diesel, and countymonthhour with those supplied by EPA for 2011. The files provided by EPA were: RegionalFuels\_2011\_20130208fuelsNMIM.zip and countymonthhour2011.zip. (This means state-supplied fuel and met data were not used in the nei2011dv1 in favor of EPA -supplied data.)
   2. The 2011 fuel tables supplied by EPA did not contain fuel data for Alaska, Hawaii, Puerto Rico, or the U.S. Virgin Islands. Data from the countyyearmonth, diesel, and gasoline tables were updated using fuel data for all years for these four states from the corresponding tables in the NCD20121201a.

# Change Log for NCD20130331

March 2013

NCD20130331, created by ERG, is a complete record of data submitted for the 2011 NEI by states via the EIS Gateway as of January 29, 2013.

The following steps describe how NCD20130331 was built.

1. This database is a copy of NCD20101201a with the following modifications:
   1. Replace the tables countyyearmonth, gasoline, diesel, and countymonthhour with those supplied by EPA for 2011. The files provided by EPA were: RegionalFuels\_2011\_20130208fuelsNMIM.zip and countymonthhour2011.zip.
   2. The 2011 fuel tables supplied by EPA did not contain fuel data for Alaska, Hawaii, Puerto Rico, or the U.S. Virgin Islands. Data from the countyyearmonth, diesel, and gasoline tables were updated using fuel data for all years for these four states from the corresponding tables in the NCD20121201a.
   3. Ensure the countymap table mapped every county to itself.
   4. Update the database using state-submitted data (submitted via EIS as of January 29, 2013):

|  |  |  |  |
| --- | --- | --- | --- |
| **Connecticut** | | | |
| **Table Name** | **Notes** | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD file. Updated entries were given a datasourceid = 9955 | | |
| **Datasource** | Inserted datasourceid 9955 for Connecticut DEEP | | |
| **Countyyearmonth** | Fuel data was updated to reflect the appropriate county assignment of RFG areas as established by EPA Region I, in conjunction with the EPA RFG group, which corresponded to assignments made in the 2002 NMIMNational County Database. | | |
| **Gasoline** | Gasoline data provided reflect survey weighted averages of 2011 RFG summer  and winter survey data. Benzene, Aromatics and Olefins are expressed as volume percent in NMIM. Winter RVP is from the latest available NCD file. Shoulder month (fall) data are an average of summer and winter months. | | |
| **External Files** | 09000.POP | | |
| **Delaware** | | | |
| **Table Name** | **Notes** | | |
| **Datasource** | Updated the date and comments fields for datasourceid 1005 | | |
| **Countyyearmonthhour** | Updated with the 2011 met data from the National Climate Data Center (NCDC) | | |
| **External Files** | 10000gc.alo | Updated based on 2008 golf course land area by county | |
| 10000hou.alo | Updated using 2011 Census Bureau’s Estimated Housing Units: 1-unit detached, 1-unit attached, and 2 units | |
| 10000log.alo | Updated using acres logged in 2006 from the Delaware Department of Agriculture, Division of Forestry | |
| 10000rvp.alo | Updated using 2008 rural land use in square kilometers, Delaware DNR | |
| 10000.pop | Recreational marine equipment population updated based on 2008 Delaware recreational boat registration data, Delaware DNR | |
| **Georgia** | | | |
| **Table Name** | **Notes** | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD files. Updated entries were given a datasourceid = 9956 | | |
| **Datasource** | Inserted datasourceid 9956 for Georgia EPD | | |
| **External Files** | 13000frm.alo | Updated defaults by replacing 2002 USDA Census of Agriculture Harvested Cropland data with 2007 data.  The USDA Census is updated every 5 years. This data is located in the U.S. Census Bureau website. | |
| 13000gc.alo | The data is extracted from the U.S. Census Bureau’s website under County Business Patterns (CBP), under table “Golf Courses and Country Clubs” (CBP NAICS 713910). The data was updated from the default database by replacing 2002 data with 2010 data. | |
| 13000com.alo | The wholesale establishment data is extracted from the U.S. Census Bureau’s website under County Business Patterns (CBP), under table “Wholesale Estabishments” (CBP NAICS 42----).  The data was updated from the default database by replacing 2002 data with 2010 data. | |
| 13000log.alo | The logging equipment data is extracted from the 2009 Timber Product Output (cu ft), Table C10 from the US Forest Service TPO database located at http://www.fia.fs.fed.us/program-features/tpo/.  The data was updated from the default database by replacing 2002 data with 2009 data. | |
| 13000lsc.alo | The data is extracted from the U.S. Census Bureau’s website under County Business Patterns (CBP), under table “Employees in Landscaping Services” (CBP NAICS 561730).  The data was updated from the default database by replacing 2002 data with 2009 data. Future plans are to use a better indicator than number of employees when data is available. Not all data was specifically provided, with some counties providing a range. As was done by EPA for the default database ((Geographic Allocation of NonRoad Engine Population Data to the State and County Level (EPA420-R-05-021, December 2005)), the midpoint of the range was used and then normalized so that all the county numbers equaled the overall state number. A further small adjustment was made to ensure that the number of employees fit within the range of bins provided. The bins consisted of number of businesses in each county with a certain range of employees (1-4, 5-9, 10-19, 20-49, 50-99, 100-249, 250-499, 500-999, 1000+). Therefore, for instance, if there were 2 businesses with 5-9 employees, there had to be at least 10 total employees listed in the dataset for that county otherwise the number was adjusted up to 10 with normalization repeated to maintain a sum equaling the state total. | |
| 13000mfg.alo | The data is extracted from the U.S. Census Bureau’s website under County Business Patterns (CBP), under a variety of tables representing varying manufacturing subsectors (wood product, paper, printing, plastics, rubber, nonmetallic mineral, metas etc.) all labeled “Number of Employees in Manufacturing” (CBP NAICS 31----, 321///, 322///, 323///, 324///, 325///, 326///, 327///, 331///,332///, 333///, 334///, 335///, 336///, 337///, 339///, and 5111//).  The data was updated from the default database by replacing 2002 data with 2010 data. Future plans are to use a better indicator than number of employees when data is available. Not all data was specifically provided, with some counties providing a range. As was done by EPA for the default database ((Geographic Allocation of NonRoad Engine Population Data to the State and County Level (EPA420-R-05-021, December 2005)), the midpoint of the range was used and then normalized so that all the county numbers equaled the overall state number. A further small adjustment was made to ensure that the number of employees fit within the range of bins provided. The bins consisted of number of businesses in each county with a certain range of employees (1-4, 5-9, 10-19, 20-49, 50-99, 100-249, 250-499, 500-999, 1000+). Therefore, for instance, if there were 2 businesses with 5-9 employees, there had to be at least 10 total employees listed in the dataset for that county otherwise the number was adjusted up to 10 with normalization repeated to maintain a sum equaling the state total. All the tables were adjusted separately and then the employee values were summed up by county. | |
| 13000oil.alo | This oil production equipment data is extracted from the U.S. Census Bureau’s website under County Business Patterns (CBP), under table “Employees in Oil & Gas Extraction, and Drilling Oil & Gas Wells (CBP NAICS 211/// and 213111).  The data was updated from the default database by replacing 2002 data with 2010 data. Not all data was specifically provided, with some counties providing a range. As was done by EPA for the default database ((Geographic Allocation of NonRoad Engine Population Data to the State and County Level (EPA420-R-05-021, December 2005)), the midpoint of the range was used and then normalized so that all the county numbers equaled the overall state number. | |
| 13000rvp.alo | This recreational vehicle parks data is extracted from the U.S. Census Bureau’s website under County Business Patterns (CBP), under table “RV (Recreational Vehicle) Parks and Campgrounds” (CBP NAICS 72121/).  The data was updated from the default database by replacing 2002 data with 2010 data. | |
| 13000pop.alo | The data is extracted from the Georgia Statistics Site at Univ of GA  http://www.georgiastats.uga.edu/sasweb/cgi-bin/broker which processes human population data by state from the U.S. Census.  The data was updated from the default database by replacing 2002 data with 2010 data. | |
| 13000hou.alo | The data was extracted from the U.S. Census data repository located at http://factfinder2.census.gov/ under table “Census H1: Housing Units Census 2010 Summary File 1”. Used this data for percent occupancy(no personal lawn equipment used in unoccupied housing) and total housing units. For percentage of total housing that is 1-2 units, used the same website under table “Selected Housing Characteristics: 2010 American Community Survey 1-Year Estimates ”. For counties not in survey used state average. In summary, the formula used (all from these to tables) is:  Single/Double Family Homes=Total Housing Units\*Fraction 1-2 units\*Fraction Occupancy | |
| **Idaho** | | | |
| **Table Name** | **Notes** | | |
| **County** | The starting and ending months and day-of-month for the ozone season were changed to match the ozone season dates defined in the August 2012 Draft Northern Ada County PM10 Maintenance Plan. | | |
| **Countyyearmonthhour** | Idaho compared the default temperature and relative humidity values to our MOVES input files. Scatter-plots shown in an Excel file submitted to EPA on the "countyyearmonthhour\_QA" tab showed a high correlation between local data used for on-road mobile modeling in MOVES and the NMIM-NONROAD defaults defined by EPA. Since Idaho had the new data, these data were updated so it would be the same as what was submitted in MOVES. | | |
| **Illinois** | | | |
| **Table Name** | **Notes** | | |
| **Countyyearmonthhour** | Used 2011 temperatures and relative humidities.   * One set of data used for northern part of state (ORD - O'Hare) * One set of data used for southern part of state (STL - Lambert St. Louis) | | |
| **Nevada** | | | |
| **Table Name** | **Notes** | | |
| **Countynrfile** | Updated to include Clark (3) and Washoe (31) counties and to direct NMIM to read the state-supplied external NONROAD files. Updated entries were given a datasourceid = 9957 | | |
| **Datasource** | Inserted datasourceid 9957 for Nevada DEQ | | |
| **Countyyearmonth** | Updated fuel data using 2011 fuel record data from the Nevada Department of Agriculture – Bureau of Petroleum Technology. | | |
| **Gasoline** | Added new gasoline ID for each county/month in 2011, updating columns for RVP and GasSulfur, and applying defaults for all other associated row values. | | |
| **External files** | 32000cen.alo | | Updated data based on the Nevada State Demographer's 2010 certified estimates. Clark (32003) and Washoe (32031) counties were also updated. September 9, 2012 downloaded Nevada State Demographer's 2010 census data Nevada County Population Projections 2010 to 2030, dated October 1, 2011. Source: http://nvdemography.org/wp-content/uploads/2011/09/2011-Projections-Email-attachment-090911.pdf |
| 32000com.alo | | Updated data to reflect number of Wholesale Establishments in every county in 2010 in Nevada using data from 2010 County Business Patterns information located at http://censtats.census.gov/cbpnaic/cbpnaic.shtml |
| 32000frm.alo | | Updated data to reflect number of Harvested Cropland in every  county in Nevada using the most recent data available from 2007 and 2002 USDA Census of Agriculture & 100 acres designated to each of the undisclosed counties. http://www.agcensus.usda.gov/Publications/2007/Full\_Report/Volume\_1,\_Chapter\_2\_County\_Level/Nevada/st32\_2\_009\_009.pdf |
| 32000gc.alo | | Updated data for the number of golf courses in every county in Nevada 2011. Information was found at http://censtats.census.gov/cbpnaic/cbpnaic.shtml, http://golf.travelnevada.com/home.aspx,  http://www.worldgolf.com/courses/usa/nevada/, http://www.nevadagolf.com/courses/, and http://www.golflink.com/golf-courses/state.aspx?state=NV |
| 32000lsc.alo | | Updated data for Nevada landscaping employees based  on the most current 2010 County Business Patterns, available in mid-2011 and located at http://censtats.census.gov/cbpnaic/cbpnaic.shtml |
| 32000oil.alo | | Updated data for the number of Employees in Oil & Gas Extraction and Drilling Oil & Gas Wells in every county in Nevada 2010/2011. Oil and gas production in Nevada is conducted in Elko, Eureka, and Nye counties. Nevada Oil Patch summary update for November/December 2011 provided totals for 2011. Updated calculations resulted in the following: according to the 2011 Oil Patch report, and the 2010 County Business Patterns (most current available) approximately 89.3% of oil production is occurring in Nye County, about 10.7% in Eureka County, and <1% in Elko County. I allocated a total of 348 employees as follows: 310 employees to Nye County, 37 employees to Eureka County, and 1 employee to Elko County.  2010 County Business Patterns (NAICS) information found at http://censtats.census.gov/cbpnaic/cbpnaic.shtml  Nevada Oil Patch – bi-monthly production report found at  http://minerals.state.nv.us/forms/ogg/oilpatch/OilPatch20111112.pdf |
| 32000rvp.alo | | Updated data for the number of RV Parks in every county in  Nevada. Information was found at various sites including http://travelnevada.com/hotels-lodging.aspx, http://www.nevadarvparks.com/,  and http://www.rv-clubs.us/nevada\_rv\_campgrounds.html |
| **New Hamshire** | | | |
| **Table Name** | **Notes** | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD file. Updated entries were given a datasourceid = 9958 | | |
| **Datasource** | Inserted datasourceid 9958 for New Hampshire DES | | |
| **External Files** | 33000.pop | | Uses registration data from the New Hampshire Division of Motor Vehicles regarding the number of outboards, inboards, and personal watercraft. |
| **North Carolina** | | | |
| **Table Name** | **Notes** | | |
| **Gasoline** | Four gasoline specifications were added to the gasoline.csv table and assigned numbers 370001, 370002, 370003, and 370004. For lack of better data, Aromatic and Olefin contents as well as E200 and E300 values were derived from the original default gasolines. | | |
| **Countyyearmonth** | Revised to use the new gasoline specifications. The diesel sulfur values were changed to expected maximums allowed in the market. The sulfur content of recreational marine diesel was selected from EPA-420-B-09-018. | | |
| **Diesel** | Diesel fuel sulfur values and ID numbers 15 and 236 were added. | | |
| **Washington** | | | |
| **Table Name** | **Notes** | | |
| **Countynrfile** | Updated to direct NMIM to read the state-supplied external NONROAD file. Updated entries were given a datasourceid = 9959 | | |
| **Datasource** | Inserted datasourceid 9959 for Washington State | | |
| **External Files** | 53000wib.alo | | Updated to reflect 2011 DOL boat registrations per county |
| 53000wob.alo | | Updated to reflect 2011 DOL boat registrations per county |
| **Wisconsin** | | | |
| **Table Name** | **Notes** | | |
| **Gasoline** | Four new gasoline ID's were added at the bottom of the table:   * 550020; winter gasoline for the six counties subject to reformulated gasoline * 550021; summer gasoline for the six counties subject to reformulated gasoline * 550040; winter gasoline for the 66 counties not subject to reformulated gasoline * 550041; summer gasoline for the 66 counties not subject to reformulated gasoline   The six counties subject to reformulated gasoline are: Kenosha (55059), Milwaukee (55079), Ozaukee (55089), Racine (55101), Washington (55131) and Waukesha (55133).  The properties for these four new gasoline ID's are the same as or are mathematically equivalent to those provided by USEPA for the MOVES County Database, dated 20120509, for the Year 2011. The properties also agree well with available gasoline survey data and federal requirements for gasoline in Wisconsin. | | |
| **Countyyearmonth** | All of the original gasoline ID's in the table (in the two columns HwyGasolineID and NRGasolineID) were replaced with the four new gasoline ID's added to the table "gasoline".  In particular:  ID 550020 replaced ID 20  ID 550021 replaced ID 21  ID 550040 replaced ID 40  ID 550041 replaced ID 41  The original ID's (20, 21, 40 and 41) were not appropriate for gasoline in Wisconsin for a variety  of reasons, including:  • Summertime RVP values were well above federally-required limits  • Sulfur values were well above federally-required limits for 2011  • A significant MTBE market share was assumed, whereas actually little to no MTBE-blended gasoline was sold in Wisconsin during 2011 | | |
| **Diesel** | One fuel ID was added: ID 15, which has a sulfur content of 15 ppm. This diesel ID is listed in the table "countyyearmonth" (in the column HwyDieselID) but was not originally included in the table "diesel". It was added to the table "diesel" for the sake of completeness and to avoid any possible error in the processing of the data. | | |

# Change Log for NCD20101201a

NCD20101201a was created by

1. copying NCD20101201

2. replacing the scctoxics table with the scctoxics table from NCD20101201Tier3.

3. copying the following tables and giving the copies an "NCD" suffix, indicating that these were the original tables. The reason for doing this is that the original tables are overwritten with custom fuels, meteorology, and representing county tables.

* Countyyearmonth
* Diesel
* Gasoline
* Countymonthhour
* Countymap

# Change Log for NCD20101201

The following states provided updates:

|  |  |  |  |
| --- | --- | --- | --- |
| **Washington DC** |  | **Changes made on 06/02/2010 update made on 10/15/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt |  | Updated the "baseyearvmt" table for 2008 based on interpolation of COG data | New Submission on 8/20 |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear |  | the file names have been changed to consistently refer to "1100108" |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  | Changed the NLEV file name |  |
|  |  |  |  |
| External Files |  | I/M cutpoints for 2008 have been added (see external file "11001208.imc"). |  |
|  |  | (IMC files were downloaded from EPA's larger NCD database, since the DC-specific NCD files did not include them.) |  |
|  |  |  |  |
|  |  | External file "1100108.imp" has been added to recognize the addition of file "11001208.imc" |  |
|  |  |  |  |
|  |  | Additional "1100108" files were added to external files to reflect file references in "countyyear" tab; no changes were made from the 05 files |  |
|  |  |  |  |
|  |  | 8/20 submission included updated 2002 & 2005 IM Files but countyyear table not updated |  |
|  |  |  |  |
| **Georgia** |  | **Changes made on 10/13/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X |  |  |
| county | X |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X |  |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files | atp13057.txt | corrected 13-county anti-tampering program |  |
|  | 1305702.reg | added Atlanta 13-county age distribution |  |
|  | 1301302.reg | added Atlanta 7-county age distribution |  |
|  | 1300102.reg | added 139-county (rest of state) age distribution |  |
|  |  |  |  |
| **Kansas** |  | **Changes made on mm/dd/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | x |  |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | x |  |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | 7 Reg Dist files |  |
|  |  | 7 Diesel fraction files |  |
|  |  |  |  |
| **Louisiana** |  | **Changes made on 10/14/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | We requested and received a spreadsheet from DOTD(Louisiana Department of Transportation and Development) with 2008 VMT for all Louisiana parishes. This spreadsheet broke the VMT down only by road type and parish. The existing NCD distribution by vehicle for each road type and county was used to further apportion the VMT by vehicle type so the table could be updated. |  |
| county | X | Ozone Season: Here it is May 1 to September 30. |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation | X | The DOTD spreadsheet had an adjustment factor for each month and road type. It was not county-specific. This was used to create the countyvmtmonthallocation table. |  |
| countyyear | X | See External Files |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | The latest distribution we have is for 2006 for the five parish area and 2005 for the rest. |  |
|  |  | These are included and the table updated. |  |
|  |  | The current Mobile 6 files for the IM program and atp are included and the table updated. |  |
|  |  |  |  |
| **Massachusetts** |  | **Changes made on 11/22/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt |  |  |  |
| county | x |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | x |  |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state | x |  |  |
|  |  |  |  |
| External Files |  | EPAMA.t2v |  |
|  |  | EPAMA.t2x |  |
|  |  | EPAMA.t2c |  |
|  |  | EPAMA.nlv |  |
|  |  | 2500008.imp |  |
|  |  | 2500008.imc |  |
|  |  | 2500003.imp |  |
|  |  | 2500005.reg |  |
|  |  |  |  |
| **Michigan** |  | **Changes made on 10/18/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | word document explains VMT calculation |  |
| county |  |  |  |
| countynrfile | X |  | Only added 26000.grw |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | pdf included justifying .reg updates |  |
| countyyearmonth | X |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline | X |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files | 26000.grw | All have date of March 16 2010 |  |
|  | 26000.pop |  |  |
|  | 26000.sea |  |  |
|  | 26000wib.alo |  |  |
|  | 26000wob.alo |  |  |
|  | 2600002.act |  |  |
|  |  |  |  |
|  |  | 83 .reg files |  |
|  |  |  |  |
| **Minnesota** |  | **Changes made on 10/15/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | The baseyearvmt table has been revised based on 2008 data from MN Department of Transportation. |  |
| county |  |  |  |
| countynrfile | X | Records have be added to the countynrfile table to link the new growth file to MN. |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | The RegDistFileName column in the countyyear table has been revised to reflect the link to the new .reg files. |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | 27000.grw |  |
|  |  | 87 .reg files |  |
|  |  |  |  |
| **Missouri** |  | **Changes made on 05/21/2010 new changes made on 10/20/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | VMT was updated for 2008 for the whole state. County level VMT was collected from the Missouri Department of Transportation. Additional county level VMT was collected from the East-West Gateway Council of Governments for the five counties currently in the St. Louis ozone non-attainment area (Franklin County, Jefferson County, St. Charles County, St. Louis County, and St. Louis City). The EPA default vehicle class and road type VMT distribution was used to allocated the county total VMT. | A review of MO Provided VMT showed huge increases (200%-300%). Further consultation with MO suggested that VMT for St Louis area should be kept and EPA default VMT used for remaining counties. This scheme used for NCD20100602. New update provided on 10/20 to be incorporated into NCD20101201 |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | The "RegDistFileName" column was filled in for 2008 for the whole state. Each county was associated with one of eight Registration Distribution files that were added as external files. |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | 2900108, 2900308, 2900908, 2901708, 2903708, 2904308, 2907108, and 2915708 are registration distribution files that have been added to as external files. All counties are associated with one of the eight distributions in the countyyear table, and the distributions list the associated counties in the beginning notes of each distribution file. |  |
|  |  |  |  |
|  |  | A list of registered vehicles was obtained from the Missouri Department of Revenue(DOR) in May of 2009. The list was grouped into EPA's 26 vehicle classes using VINPower, a VIN decoding software. The software was run by ESP Solutions Inc., a private contractor and developer of the VINPower software. The small portion of vehicles not assigned to a vehicle class by the VINPower software were grouped by the Missouri DNR staff using the vehicle make and model information from the Missouri DOR's registration list. |  |
|  |  |  |  |
|  |  | Four distributions were created for the five proposed ozone non-attainment areas (Perry county and Ste. Genevieve county were grouped into a single distribution). |  |
|  |  |  |  |
|  |  | Four distributions were create for the remaining counties of the state. These counties were grouped in allignment with our regional office county groups. |  |
|  |  |  |  |
| **Nevada** |  | **Changes made on 10/26/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | See Text File for documentation (plus references link) |  |
| county |  |  |  |
| countynrfile | X | See Text File for documentation (plus references link) |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | I updated the Registration distribution files (i.e. the RegDistFileName column) by obtaining DMV Registration Distribution Data. |  |
| countyyearmonth | X | I updated the HwyDieselID, HwyGasolineID, NRGasolineID, and NRDieselID columns based on information provided by the Nevada Department of Agriculture –  Bureau of Petroleum Technology (BPT). The BPT does not have Sulfur Content data for Recreational Marine diesel. |  |
| countyyearmonthhour |  |  |  |
| diesel | X | I updated the HwyDieselID Column with the data provided to me by the BPT. The BPT provided me with 2008 Sulfur Content data for Highway diesel.  Unfortunately, to update the CountyYearMonth NCD Table you need Sulfur Content data for every Month in every County. The BPT was unable to provide a complete set of Sulfur Content data. Therefore, I interpolated the Sulfur Content data using instructions from an email from Harvey Michaels |  |
| gasoline | X | The HwyGasolineID/NRGasolineID Columns contains gasoline IDs that correspond to different gasoline properties in the Gasoline NCD Table. I updated the Gasoline NCD Table with the data provided to me by the BPT. The BPT provided me with 2008 Sulfur Content data for Gasoline and 2008 Reid Vapor Pressure (RVP) data. Unfortunately, to update the CountyYearMonth NCD Table you need these gasoline properties for every Month in every County. The BPT was unable to provide a complete set of these gasoline properties. Therefore, I interpolated these gasoline properties according to Section I of Pechan, 2009 and the Supplemental Instructions for Interpolating Gasoline Properties2.doc. |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | 10 Nonroad Files |  |
|  |  | 15 .reg files |  |
|  |  | 3200099.imp |  |
|  |  | atp32003.txt |  |
|  |  |  |  |
| **Ohio** |  | **Changes made on 11/08/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | x | 4. VMT data updated using Ohio DOT's 2008 VMT data.  4-1. This data was selected to be the most reasonable VMT information for Ohio because Traffic Demand Model and HPMS data are utilized to produce.  4-2. This data also incorporated MPOs VMT estimate. 4-4. For more details, please see the attached document, "Ohio\_VMT\_Analysis\_OEPA.docx". |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | x | 3. Registration data have been updated with ERG's VIN Decoding data.  5. I/M Programs files were added for Cleveland, Cincinnati, and Dayton areas based on Ohio DOT and U.S. EPA data as following: Cleveland area (103, 85, 93, 133, 35, 55, 153): 3903503.imp Cincinnati area (25, 165, 61, 17): 3906103.imp Dayton area (57, 23, 113): 3911303.imp |  |
| countyyearmonth | x | Different gasolines have been assigned for eight counties (17, 23, 25, 57, 61, 109, 113 and 165) based on state recommendations data. |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files | 39000.grw |  |  |
|  |  | 4 atp files |  |
|  |  | 5 imp files |  |
|  |  | 89 reg files |  |
|  |  |  |  |
| **South Carolina** |  | **Changes made on 10/26/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | x | We updated the VMT to match the 2008 VMT data received from our S.C. DOT. |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear |  |  |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  |  |  |
|  |  |  |  |
| **Nashville TN** |  | **Changes made on 11/15/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | x | The DVMT that we received from the Tenn. Dept. of Transportation are as follows:   Interstate   Rural 801,879  Urban 8,378,462 Expressway  Urban 1,413,160 Arterial   Rural 553,259  Urban 6,396,165 Collector   Rural 165,096  Urban 921,378 Local   Rural 105,259  Urban 2,739,761 Total 21,474,419   The data is probably shifted more to LDGV due to the use of the use of the 2008 VMT Fractions for 16 Vehicle Classes from MOBILE6 Manual, 5.3.2, pg 219; Constants from Appendix D, which puts LDV at around 67 to 78 percent of VMT for the different road types.   The changes appear to be appropriate due to using the VMT fraction constants and the DVMT provided by TDOT. |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear |  |  |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  |  |  |
|  |  |  |  |
| **Wisconsin** |  | **Changes made on 11/08/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | x |  |  |
| county |  |  |  |
| countynrfile | x |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | x |  |  |
| countyyearmonth | x |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline | x |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | 6 Nonroad files including 55000.GRW |  |
|  |  | 6 .imp |  |
|  |  | 5500002.REG |  |
|  |  | Avg Speed Dist (7 counties) 126 files |  |

# Change Log for NCD20100602

The following states provided updates:

|  |  |  |  |
| --- | --- | --- | --- |
| **Arkansas** |  | **Changes made on 05/11/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | updated with VMT data from Arkansas Highway and Transportation Department |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | MOBILE6 Vehicle Registration files for 75 counties provided from Arkansas Highway and Transporatation Department. | JVB made assignments because AR did not provide. |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | 75 .reg files provided |  |
|  |  |  |  |
| **Washington DC** |  | **Changes made on 06/02/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt |  | Updated the "baseyearvmt" table for 2008 based on interpolation of COG data |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear |  | the file names have been changed to consistently refer to "1100108" |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  | Changed the NLEV file name |  |
|  |  |  |  |
| External Files |  | I/M cutpoints for 2008 have been added (see external file "11001208.imc"). |  |
|  |  | (IMC files were downloaded from EPA's larger NCD database, since the DC-specific NCD files did not include them.) |  |
|  |  |  |  |
|  |  | External file "1100108.imp" has been added to recognize the addition of file "11001208.imc" |  |
|  |  |  |  |
|  |  | Additional "1100108" files were added to external files to reflect file references in "countyyear" tab; no changes were made from the 05 files |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Hawaii** |  | **Changes made on 05/27/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | 2008 vmt at the county level of detail obtained from Hawaii Department of Transportation were distributed in the same proportions as the default 2008 vmt by county, vehicle type, and road type.   Our updated baseyearvmt.csv table does not include data for Kalawao county as we were not able to obtain specific vmt data for this county. We also did not include the default EPA data for this county as we did not feel that it was representative of Kalawao (we believe the default data is too large) for the following reasons: Kalawao county is located on the Kalaupapa Peninsula on the island of Molokai. This small penininsula of Kalaupapa is isolated from the rest of Molokai by sea cliffs over a quater-mile high where the only land access is by mule trail. The state once quarantined persons suffering from leprosy (Hansens disease) to Kalaupapa beginning in the 1860s. Currently, no new patients or other permanent residents are admitted and visitors are only permitted as part of officially santioned tours. The population of Kalawao in 2008 was 117. The county land area is only 13 square miles (the town being approximately 0.5 square miles and distance across the peninsula being less than 2.5 miles) and it is ranked by medium household income as the poorest county in the United States. |  |
| county | X | Revised OzoneSeasonStartMonth from 6 to 1 Revised OzoneSeasonEndMonth from 8 to 12  as Hawaii's ozone monitoring season is year round (40 CFR 58, App. D, Table D-3). |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear |  |  |  |
| countyyearmonth | X | Revised Hawaii HwyGasoline and NRGasoline IDs from: 2504 and 2506 to 150001 |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline | X | For Ids 2504 and 2505, RVP is specified as 13.0886 (winter) and 9.39714 (summer), respectively. Unlike other areas where RVP varies in the summer and winter periods, Hawaii's retailers cannot sell gasoline with an RVP greater than 11.5 psi without violating standards. To reflect this discrepancy, added new Id 150001 where RVP was revised to 9.61 (based on average of various local data) to reflect Hawaii conditions. Also revised other gasoline parameters: GasolineSulfur, AromaticContent, OlefinContent, BenzeneContent, E200 and E300 (based on averages of various local data). |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  |  |  |
|  |  |  |  |
| **Maine** |  | **Changes made on 04/22/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | Maine DEP is supplying actual 2008 VMT, as provided by the Maine Dept. of Transporation, August 2009. | Not all vehicle class/road type combinations included in ME submittal. Email on 4/26 confirmed missing records should be zero VMT |
| county | X | After checking with our Meterological staff, I have been informed that ozone season is now considered April 1 through September 30 of each year. This table has been updated to reflect that. No other changes were made to the Stage 2 inputs. |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | Maine DEP has reviewed the .atp and .imp files, renamed them for the 2008 inventory, and made the appropriate file name changes to countyyear.csv. |  |
| countyyearmonth | X | Maine DEP has added two summer gasoline profiles and amended the table as appropriate. |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline | X | Maine DEP has added two summer gasoline profiles, 238806 and 238812. These profiles reflect Maine law which prohibits the sale of gasoline in the summer with an RVP of greater than 7.8 in the southern counties and 9.0 in the northern counties. |  |
| naturalgas |  |  |  |
| state | X | This table was amended to include the appropriate extensions to the file names. |  |
|  |  |  |  |
| External Files |  | 7 Files in External Files Folder |  |
|  |  |  |  |
| **Maryland** |  | **Changes made on 05/25/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | Updated VMT column based on 2008 SHA traffic data. A total of 8064 rows of records have been updated. |  |
| county | X | barometric pressure values were obtained from the airports in the State of MD. Counties were assigned to each airport through what we designated as airport mapping. For instance, the Baltimore Area (comprising of counties of Anne Arundel, Baltimore, Carroll, Harford, Howard and the City of Baltimore) was assigned to the BWI Airport. A total of 24 rows of records have been updated. |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation | X | Updated VMT allocation factors based on 2008 SHA traffic data. For certain county/roadtype/vehicle type combinations, EPA default data was used where data was not avilable. A total of 55296 rows of records have been updated. |  |
| countyyear | X | Updated the external file names for ATP, IM, RegDist, VMTbyHour and DieselFraction based on the latest I/M data and motor vehicle registration data and the latest planning assumptions. A total of 24 rows of records have been updated. |  |
| countyyearmonth | X | Updated HwyDieselID and HwyGasolineID based on data based on the latest planning assumptions. A total of 288 rows of records have been updated. | After Consultation with OTAQ, Pechan will use HwyGasoline in the NRGasoline Column as well |
| countyyearmonthhour | X | Updated the table with the 2008 avg. hourly temperature and relative humidity. The temperature and relative humidity values were obtained from the airports in the State and counties were assigned to each airport through what we designated as airport mapping. For instance, the Baltimore area (Anne Arundel, Baltimore, Carroll, Harford, Howard and Baltimore City) was assigned to BWI Airport. | After consultation with Otaq, Pechan will not update this table because procedure MD used not as sophisticated as EPA's procedure. |
| diesel | X | Updated this table with the 2008 avg. monthly diesel values obtained from the MD fuel data analysis. A total of nine rows of records have been added. |  |
| gasoline | X | Added new gasolineID with updated gasoline fuel properties based on data obtained from the MD fuel data analysis. A total of 24 rows of records have been added (gasolineID 240001 to 240024). Even though the ethanol content in RFG counties ranged between 11.2 and 11.7, this data element was reduced to 10.6 based on the Production 2 submission feedback report of 5/17/2010. |  |
| naturalgas |  |  |  |
| state | X |  |  |
|  |  |  |  |
| External Files |  | II)The following changes have been made to the external files from National Mobile Inventory Model |  |
|  |  | ( NMIM) National County Database (NCD)for the State Maryland for the Year 2008: |  |
|  |  |  |  |
|  |  | 1. Average Speed Distributions (\*\*\*\*\*YY.fwX and \*\*\*\*\*YY.arX files): Updated average speed |  |
|  |  | distribution files based on 2008 SHA traffic data. Converted average speed distribution files |  |
|  |  | in M6 format to NCD-referenced external file format with file names complying to NCD naming |  |
|  |  | convention and also following a mapping of functional class and vehicle types between M6 and |  |
|  |  | NEI formats. A total of 18 NCD external files were prepared for each county, including 9 files |  |
|  |  | with fwX extension for freeways and 9 files with arX extension for arterials. Each NCD external |  |
|  |  | file stores VMT distribution across 14 average speed ranges for each of the 24 hours of the day. |  |
|  |  |  |  |
|  |  | 2. VMT by Hour (\*\*\*\*\*YY.vmt files): Included VMT by hour files based on 2008 SHA traffic data. The |  |
|  |  | hourly VMT distribution by facility type in M6 format for each county was processed to create a |  |
|  |  | county-average distribution to be used as NCD-referenced external files. Each NCD external file |  |
|  |  | contains the fraction of VMT that occurs at each hour of the day for all facility types in the |  |
|  |  | applicable county. |  |
|  |  |  |  |
|  |  | 3. Diesel Fractions (\*\*\*\*\*YY.dsf files): Updated diesel fraction files prepared from the 2008 motor |  |
|  |  | vehicle registration data obtained from Maryland's Motor Vehicle Adminstration as the latest |  |
|  |  | planning assumptions. |  |
|  |  |  |  |
|  |  | 4. Anti-tampering Programs (\*\*\*\*\*YY.atp files): Updated anti-tampering program files based on the |  |
|  |  | ATP design values. |  |
|  |  |  |  |
|  |  | 5. IM Programs (\*\*\*\*\*YY.imp files): Updated IM program files based on the IM Program design values |  |
|  |  | and IMdata collected at the Maryland's Vehicle Emission Inspection stations. |  |
|  |  |  |  |
|  |  | 6. IM Cutpoint (\*\*\*\*\*NYY.imc files): Updated the IM CutPoint files based on the lM Program design |  |
|  |  | values. |  |
|  |  |  |  |
|  |  | 7. Distribution of Vehicle Registrations (\*\*\*\*\*YY.reg files): Updated the motor vehicle registration |  |
|  |  | distribution prepared from the 2008 motor vehicle registration data obtained from Maryland's Motor |  |
|  |  | Vehicle Adminstration as the latest planning assumptions. |  |
|  |  |  |  |
| **Missouri** |  | **Changes made on 05/21/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | VMT was updated for 2008 for the whole state. County level VMT was collected from the Missouri Department of Transportation. Additional county level VMT was collected from the East-West Gateway Council of Governments for the five counties currently in the St. Louis ozone non-attainment area (Franklin County, Jefferson County, St. Charles County, St. Louis County, and St. Louis City). The EPA default vehicle class and road type VMT distribution was used to allocated the county total VMT. | QA of VMT showed extreme increases (doubling and tripling of VMT) so after consultation with MO, EPA default data was used for most of the state, and MO VMT used for the St Louis counties. |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | The "RegDistFileName" column was filled in for 2008 for the whole state. Each county was associated with one of eight Registration Distribution files that were added as external files. |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | 2900108, 2900308, 2900908, 2901708, 2903708, 2904308, 2907108, and 2915708 are registration distribution files that have been added to as external files. All counties are associated with one of the eight distributions in the countyyear table, and the distributions list the associated counties in the beginning notes of each distribution file. |  |
|  |  |  |  |
|  |  | A list of registered vehicles was obtained from the Missouri Department of Revenue(DOR) in May of 2009. The list was grouped into EPA's 26 vehicle classes using VINPower, a VIN decoding software. The software was run by ESP Solutions Inc., a private contractor and developer of the VINPower software. The small portion of vehicles not assigned to a vehicle class by the VINPower software were grouped by the Missouri DNR staff using the vehicle make and model information from the Missouri DOR's registration list. |  |
|  |  |  |  |
|  |  | Four distributions were created for the five proposed ozone non-attainment areas (Perry county and Ste. Genevieve county were grouped into a single distribution). |  |
|  |  |  |  |
|  |  | Four distributions were create for the remaining counties of the state. These counties were grouped in allignment with our regional office county groups. |  |
|  |  |  |  |
| **North Carolina** |  | **Changes made on 05/06/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | Updated VMT (millions) disaggregated to 12 road types and 28 vehicle types for all 100 counties. | QA of VMT showed large increases. Consultation with NC revealed the use of the travel demand model for some metro counties, and in counties going to travel demand model in the near future used HPMS\*1.3 QA also found an error in VMT incorrectly assigned to 4 counties starting with the letter M. Corrected in an August submission |
| county |  | Updated Ozone season from 5/1 to 9/30 as input to this table. |  |
| countynrfile |  | NMIM NONROAD seasonal activity default setting is changed from Mid Atlantic to South East due to the fact that the climatic and meteorological patterns of North Carolina are more inline with the South East States rather than to the Mid Atlantic Region States. This change is applied to all 100 counties and is reflected in the external file 37000.sea which is provided in the external file folder, and is shown in several columns of this Countynrfile Table. |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | \* One Anti-Tampering Program for all 100 counties using 3700191.atp as representing external file  \* I/M program of 7 counties representing total 48 counties as specified in this table  \* VMT by Speed for 100 counties (total of 1,800 external files) are included in the external file folder.  \* Registration distribution for Durham, Forsyth, Guilford, Gaston, Mecklenburg, Orange, Wake counties,rest of NC counties (93) use 3700108.reg; Gaston/Mecklenburg share one registration distribution.  \* Hourly VMT distribution is unique to Cabarrus, Durham, Gaston, Lincoln, Mecklenburg, Orange,Rowan, Stanly, Union, & Wake counties. Remaining NC counties use a flat hourly VMT profile file (3700108.vmt).  \* One NONROAD seasonality file for all 100 counties as indicated in countynrfile above. | Problem with AvgSpeedVMT files in Avery (37011) County. Working to resolve |
| countyyearmonth | X | North Carolina County Specific monthly Reid Vapor Pressure (RVP) values are updated in this table with newly assigned Gasoline ID while keeping all other default fuel parameters in the table the samme. There are only two scenarios which cover all 100 counties. |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline | X | New Gasoline IDs are inserted to this table from countyyearmonth above to reflect NC county specific gasoline RVP. |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | Anti-Tampering files (.atp): 1 |  |
|  |  | I/M Files (.IMP): 7 |  |
|  |  | Hourly VMT distribution files (.VMT): 11 |  |
|  |  | VMT by Speed Distribution files (AR\*, FW\*): 1,800 |  |
|  |  | Registration Files (.reg): 7 |  |
|  |  | NONROAD Season activity data files (.sea): 1 |  |
|  |  |  |  |
| **New Hampshire** |  | **Changes made on 04/22/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | Used NH-specific VMT provided by NHDOT. Apportioned it by road type and vehicle type using EPA's baseyearvmt.csv file to calculate proportions. |  |
| county | X | Changed Stage 2 start year from 1998 to 1993. |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | Changed this table to reference external files atp33000.txt and 3300000.imp, which contain NH-specific ATP and I/M inputs. |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  | Made sure that the 33000.nlv file is equal to our nlevne.d file. It matched, so no changes made to this table. |  |
|  |  |  |  |
| External Files |  |  |  |
|  |  |  |  |
| **Pennsylvania** |  | **Changes made on 06/24/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | Updated VMT column based on 2008 RMS traffic data using the PPSuite process. 2008 monthly VMT by 28 M6-vehicle type/facility type/county was processed to obtain annual VMT in NCD table format (by vehicle type/road type/county). A total of 22512 rows of records have been updated. |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation | X | Updated VMT allocation factors based on 2008 RMS traffic data using PPsuite process. 2008 monthly VMT by 28 M6-vehicle-type/facility type/county was used to develop VMT allocation factors in NCD table format (by county/vehcile type/road type/month). A total of 154368 rows of records have been updated. |  |
| countyyear | X | Updated the external file names for ATP, IM, RegDist, VMTbyHour and AvgSpeedDistBase based on data provided by PENNDOT as latest planning assumptions. Deleted the existing entries in the datasourceID field. A total of 67 rows of records have been updated. |  |
| countyyearmonth | X | Updated HwyGasolineID based on data provided by PENNDOT as latest planning assumptions. A total of 804 rows of records have been updated. |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline | X | Added new gasolineID with updated gasoline fuel properties based on data provided by PENNDOT as latest planning assumptions. A total of 14 rows of records have been added (gasolineID 420001 to 420014). |  |
| naturalgas |  |  |  |
| state | X | Updated the external file names for NLEV, T2ExhPhaseIn, T2EvapPhaseIn and T2Cert fields based on file naming convention. Deleted the existing entry in the datasourceID field. |  |
|  |  |  |  |
| External Files |  | 1. Average Speed Distributions (\*\*\*\*\*YY.fwX and \*\*\*\*\*YY.arX files): Updated average speed distribution files based on 2008 RMS traffic data. Converted average speed distribution files in M6 format to NCD-referenced external file format with file names complying to NCD naming convention and also following a mapping of functional class and vehicle types between M6 and NEI formats. A total of 18 NCD external files were prepared for each county, including 9 files with fwX extension for freeways and 9 files with arX extension for arterials. Each NCD external file stores VMT distribution across 14 average speed ranges for each of the 24 hours of the day. |  |
|  |  |  |  |
|  |  | 2. VMT by Hour (\*\*\*\*\*YY.vmt files): Included VMT by hour files based on 2008 RMS traffic data. The hourly VMT distribution by facility type in M6 format for each county was processed to create a county-average distribution to be used as NCD-referenced external files. Each NCD external file contains the fraction of VMT that occurs at each hour of the day for all facility types in the applicable county. |  |
|  |  |  |  |
|  |  | 3. Anti-tampering Programs (\*\*\*\*\*YY.atp files): Updated anti-tampering program files. |  |
|  |  |  |  |
|  |  | 4. IM Programs (\*\*\*\*\*YY.imp files): Updated IM program files. |  |
|  |  |  |  |
|  |  | 5. Distribution of Vehicle Registrations (\*\*\*\*\*YY.reg files): Updated vehicle registration distribution files based on 2008 registration data as provided by PENNDOT as latest planning assumptions. |  |
|  |  |  |  |
|  |  | 6. NLEV Program (42000.nlv): updated file name based on file naming convention. |  |
|  |  |  |  |
|  |  | 7. Tier2 Exhaust Emission Standard Phase-In Program (42000.t2x): updated file to include CALLEVII 08 Phase-in. |  |
|  |  |  |  |
|  |  | 8. Tier2 Evaporative Emission Standard Phase-In Program (42000.t2v): updated file to include CALLEVII 08 Phase-in. |  |
|  |  |  |  |
|  |  | 9. AlternateTier2 Exhaust Emission Standards (42000.t2c): updated file to include mobile6.2 defaults. |  |
|  |  |  |  |
| **Vermont** |  | **Changes made on 04/22/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | Entered county specific VMT by VClass by Road type into "baseyearvmt.csv" file. VMT in file now add up to total VMT estimated by VT Agency of Transportation for 2008. |  |
| county |  |  |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | Removed "5000002.reg" file from external files, and deleted it from the "RegDistFileName" field of the "countyyear.csv" file. Registration data is outdated, and VT DEC will use default EPA registration data for 2008, as better data currently are lacking. | External file was not actually deleted, only the countyyear table was updated |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Virginia** |  | **Changes made on 05/26/2010** |  |
|  |  |  |  |
| **Table Name** | **Revised** | **State Comments** | **Pechan Comments** |
| baseyearvmt | X | The base year VMT table was updated using calendar year 2008 traffic data obtained from Virginia DOT. The EPA NCD for Virginia was downloaded from the EPA ftp website (ftp://ftp.epa.gov/EmisInventory/2008\_nei/ncd\_files). The EPA preliminary 2008 base year VMT table from the ftp site was used to develop vehicle class allocation factors for the MOBILE6 28 vehicle classes. The derived vehicle class allocation factors were applied to the VMT by county by road type provided by VDOT in order to determine county-level VMT by MOBILE6 vehicle class and road type. |  |
| county | X | The county table was reviewed and the stage 2 percentages were updated to reflect the current activity in Virginia based on the EPA guidance document Procedures for Emission Inventory Preparation Volume IV: Mobile Sources, EPA 420-R-92-009.   Corrections/modifications were made for columns: LDVStage2Percent, HDVStage2Percent, Stage2StartYear. |  |
| countynrfile |  |  |  |
| countyvmtmonthallocation |  |  |  |
| countyyear | X | The county table was reviewed and several items were updated. The vehicle registration distribution was updated for each jurisdiction based on 2008 data obtained from the Virginia Department of Motor Vehicles. The I/M file input was updated based on 2008 program details. The speed distribution input files were updated based on average road speed data obtained from Virginia DOT.   Corrections/modifications were made for columns: IMFileName, RegDistFileName, DieselFractFileName, AvgSpeedDistBaseFileName, ATPFileName. |  |
| countyyearmonth |  |  |  |
| countyyearmonthhour |  |  |  |
| diesel |  |  |  |
| gasoline |  |  |  |
| naturalgas |  |  |  |
| state |  |  |  |
|  |  |  |  |
| External Files |  | The “External Files” folder in the 51000.zip file contains updated \*.reg files for all Virginia jurisdictions, two updated \*.imp files for the Virginia IM program, and numerous updated \*.ar\* and \*.fw\* files prepared using new average road speed data provided by VDOT. |  |

**VMT Upates:**

Maricopa County AZ, Jefferson County KY, Idaho, Illinois, and Texas submitted SCC level data to the EIS. Allocation to the NMIM level (28 Vehicle Classes, 12 Road Types) from the SCC level (12 Vehicle Classes, 12 Road Types) was done by summing a previous set of VMT data at the NMIM level to the SCC level, then developing allocations based on the ratio of NMIM to SCC VMT.

Maricopa AZ and Jefferson KY used VMT from baseyearvmt where baseyear = 2002

Idaho used VMT from baseyearvmt where baseyear = 2005

This procedure worked for cases where there is nonzero VMT in all 144 SCC combinations, and in cases where the zero VMT in a previous year is still zero VMT in the subsequent year. Texas and Illinois had cases where an SCC in a previous year had zero VMT, but in the current submission had nonzero VMT. Using the previously described procedure resulted in state supplied VMT being zeroed out for the cases described about. So Pechan then reverted to using MOBILE6 default allocations.

Illinois also had an additional step. Under previous years, IL was able to report HDDV data using the more generic SCC 2230070*nnn* instead of their individual SCCs 2230071-2230075. The EIS did not allow these generic SCCs, so IL submitted using 2230071 (2BHDDV) as representative for all HDDV. So an additional allocation step involved summing MOBILE6 default SCCs 71-75 (to get HDDV total) together (keeping road types separate) then allocating IL’s 71 by the ratio of default 71 (and 72-75) to default HDDV total

A new 2008 baseyearvmt was developed using the same procedure Pechan has used for previous years which is based on HPMS data and MOBILE6 default allocations. This table was used for all areas not previously listed.

After this table was developed, it was decided that if a State/Local Agency supplied VMT in 2002 or 2005, this data should be used for allocation, rather than the MOBILE6 default allocation used.

This was handled by summing new 2008 VMT to the State level (one VMT for each state). 2002 (or 2005) baseyearvmt data was also summed to the state total (again, one VMT per state). The statewide allocation table created by ratio of 2002 (or 2005) baseyearvmt record to the baseyearvmt state total (#counties\*336 records). 2008 VMT multiplied by the allocation table produced an updated baseyearVMT.

The following areas used 2002 data for allocation of 2008 HPMS data: AL, MA, MS, NJ, NY, OR, RI, WA, Lancaster NE, and Clark NV.

The following areas used 2005 data for allocation of 2008 HPMS data: CA, CO, DE, GA, MI, SC, TN, UT, WV, and NV except for Clark and Washoe counties.

This leaves the following areas using MOBILE6 default allocation for 2008 HPMS data:AK, AZ (except Maricopa), CT, FL, IN, IA, KS, KY(except Jefferson), LA, MN, MT, NE (except Lancaster), Washoe NV, NM, ND, OH, OK, SD, WI, WY, PR, VI.

The following table summarizes allocation method. SCC in the 2008 column denotes SCC VMT data submitted. State in the allocation Year column denotes state supplied VMT; State/default for IL and TX explained above.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VMT Submitted | | |  |  |
| Area | 2002 | 2005 | 2008 |  | Allocation Year |
| AL | X |  |  |  | 2002 |
| AK |  |  |  |  | default |
| AZ |  |  |  |  | default |
| AZ Maricopa | X |  | SCC |  | State |
| AR | X |  | X |  | State |
| CA | X | X |  |  | 2005 |
| CO | X | X |  |  | 2005 |
| CT |  |  |  |  | default |
| DE | X | X |  |  | 2005 |
| DC | X |  | X |  | State |
| FL |  |  |  |  | default |
| GA | X | X |  |  | 2005 |
| HI |  | X | X |  | State |
| ID | X | X | SCC |  | State |
| IL | X |  | SCC |  | State/default |
| IN |  |  |  |  | default |
| IA |  |  |  |  | default |
| KS |  |  |  |  | default |
| KY |  |  |  |  | default |
| KY Jefferson | X |  | SCC |  | State |
| LA |  |  |  |  | default |
| ME | X | X | X |  | State |
| MD | X | X | X |  | State |
| MA | X |  |  |  | 2002 |
| MI | X | X |  |  | 2005 |
| MN |  |  |  |  | default |
| MS | X |  |  |  | 2002 |
| MO | X |  | X |  | State |
| MT |  |  |  |  | default |
| NE |  |  |  |  | default |
| NE Lancaster | X |  |  |  | 2002 |
| NV | X except Washoe | X except Clark and Washoe |  |  | 2005 |
| NV Clark | X |  |  |  | 2002 |
| NV Washoe |  |  |  |  | default |
| NH |  |  | X |  | State |
| NJ | X |  |  |  | 2002 |
| NM |  |  |  |  | default |
| NY | X |  |  |  | 2002 |
| NC | X | X | X |  | State |
| ND |  |  |  |  | default |
| OH |  |  |  |  | default |
| OK |  |  |  |  | default |
| OR | X |  |  |  | 2002 |
| PA | X |  | X |  | State |
| RI | X |  |  |  | 2002 |
| SC |  | X |  |  | 2005 |
| SD |  |  |  |  | default |
| TN | X | X |  |  | 2005 |
| TX | X | X | SCC |  | State/default |
| UT | X | X |  |  | 2005 |
| VT | X |  | X |  | State |
| VA | X | X | X |  | State |
| WA | X |  |  |  | 2002 |
| WV | X | X |  |  | 2005 |
| WI |  |  |  |  | default |
| WY |  |  |  |  | default |
| PR |  |  |  |  | default |
| VI |  |  |  |  | default |

# Change Log for NCD20091016

All of the emission inventory system (EIS) checks were applied to the default database. These changes to the database were required (a-d) in order to pass the checks:

**A) The sum of ETBE Mkt Share, ETOH Mkt Share, MTBE Mkt Share, and TAME Mkt Share must be less than or equal to 1.**

**Required the following mySql code:**

-- Check 133.760 -- MktShare Calculation

Drop table if exists gasolineFixed;

CREATE TABLE `gasolineFixed` (

`RVP` float NOT NULL DEFAULT '0',

`GasSulfur` float NOT NULL DEFAULT '0',

`GasMaxSulfur` float NOT NULL DEFAULT '0',

`RVPOxyWaiver` tinyint(4) NOT NULL DEFAULT '1',

`ETOHVolume` float NOT NULL DEFAULT '0',

`ETOHMktShare` float NOT NULL DEFAULT '0',

`MTBEVolume` float NOT NULL DEFAULT '0',

`MTBEMktShare` float NOT NULL DEFAULT '0',

`ETBEVolume` float NOT NULL DEFAULT '0',

`ETBEMktShare` float NOT NULL DEFAULT '0',

`TAMEVolume` float NOT NULL DEFAULT '0',

`TAMEMktShare` float NOT NULL DEFAULT '0',

`AromaticContent` float NOT NULL DEFAULT '0',

`OlefinContent` float NOT NULL DEFAULT '0',

`BenzeneContent` float NOT NULL DEFAULT '0',

`E200` float NOT NULL DEFAULT '0',

`E300` float NOT NULL DEFAULT '0',

`RFG` char(1) NOT NULL DEFAULT '',

`Gasolineid` smallint(6) NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`Gasolineid`),

KEY `index10` (`RVP`)

) ENGINE=MyISAM AUTO\_INCREMENT=4462 DEFAULT CHARSET=latin1 ;

Insert into GasolineFixed

select \*

-- ETBEMktShare + ETOHMktShare + TAMEMktShare + MTBEMktShare as Total,

-- GasSulfur as Factor

from gasoline;

Alter Table GasolineFixed add Total double;

Alter Table GasolineFixed add Factor double;

Alter table GasolineFixed add OkOrNot float;

Update gasolineFixed

set Total = ETBEMktShare + ETOHMktShare + TAMEMktShare + MTBEMktShare;

Update GasolineFixed

Set OkOrNot = CASE

when total = 0.0 then 1

when total >= 0.0 and total < 1.0 then 2

when total = 1.0 then 3

when total > 1.0 and total < 1.1 then 4

else 5

END ;

Select OkOrNot,

Count(\*)

From GasolineFixed

Group by OkOrNot

Order by OkOrNot; -- (1, 213;

-- 2, 3115;

-- 3, 795;

-- 4, 338)

Select min(total), max(total) from GasolineFixed where OkOrNot = 1; -- 0, 0

Select min(total), max(total) from GasolineFixed where OkOrNot = 2; -- .003700, .9999

Select avg(total) from GasolineFixed where OkOrNot = 2; -- .736 ...

Select min(total), max(total) from GasolineFixed where OkOrNot = 3; -- 1, 1

Select min(total), max(total) from GasolineFixed where OkOrNot = 4; -- 1.000009, 1.0405599

Select avg(total) from GasolineFixed where OkOrNot = 4; -- 1.00367

Select min(total), max(total) from GasolineFixed where OkOrNot = 5; -- null, null

Update GasolineFixed set Factor = 0.0;

Update GasolineFixed

Set Factor = 1.0 / Total

Where Total > 0.0;

Update GasolineFixed set ETBEMktShare = ETBEMktShare \* Factor Where Factor > 0.0 and Factor < 1.0;

Update GasolineFixed set ETOHMktShare = ETOHMktShare \* Factor Where Factor > 0.0 and Factor < 1.0;

Update GasolineFixed set TAMEMktShare = TAMEMktShare \* Factor Where Factor > 0.0 and Factor < 1.0;

Update GasolineFixed set MTBEMktShare = MTBEMktShare \* Factor Where Factor > 0.0 and Factor < 1.0;

Update GasolineFixed

Set Total = ETBEMktShare + ETOHMktShare + TAMEMktShare + MTBEMktShare

Where Factor > 0.0 and Factor < 1.0;

Update GasolineFixed

Set OkOrNot = CASE

when total = 0.0 then 1

when total >= 0.0 and total < 1.0 then 2

when total = 1.0 then 3

when total > 1.0 and total < 1.1 then 4

else 5

END ;

Select OkOrNot,

Count(\*)

From GasolineFixed

Group by OkOrNot

Order by OkOrNot; -- (1 213; Old

-- 2 3115; Old

-- 3 795; Old

-- 4 338) Old

-- 1 213 New

-- 2 3177 New

-- 3 795 New

-- 4 276 New

Select min(total), max(total) from GasolineFixed where OkOrNot = 1; -- same

Select min(total), max(total) from GasolineFixed where OkOrNot = 2; -- same

Select avg(total) from GasolineFixed where OkOrNot = 2; -- Old: .736, New: .741

Select min(total), max(total) from GasolineFixed where OkOrNot = 3; -- same

Select min(total), max(total) from GasolineFixed where OkOrNot = 4; -- 1.000000006, 1.000000031665

Select avg(total) from GasolineFixed where OkOrNot = 4; -- Old: 1.00367, New: 1.000000170

Select min(total), max(total) from GasolineFixed where OkOrNot = 5; -- same

Alter Table GasolineFixed drop Total;

Alter Table GasolineFixed drop Factor;

Alter table GasolineFixed drop OkOrNot;

**B) If either TAME Mkt Share or TAME Volume is zero, the other must also be 0.**

# -- Check 137.764 --TAMEMktShare and TAMEVolume

# Select count(\*) From gasolineFixed Where TAMEMktShare <> 0.0 and TAMEVolume = 0.0; -- ( 0)

# Select count(\*) From gasolineFixed Where TAMEMktShare = 0.0 and TAMEVolume <> 0.0; -- ( 16)

# Select count(\*) From gasolineFixed Where TAMEMktShare = 0.0 and TAMEVolume = 0.0; -- (3263)

# Select count(\*) From gasolineFixed Where TAMEMktShare <> 0.0 and TAMEVolume <> 0.0; -- (1182)

# Select count(\*) from gasolineFixed; -- (4461)

# Update gasolineFixed

# Set TAMEVolume = 0.0

# Where TAMEMktShare = 0.0 and TAMEVolume <> 0.0;

# Select count(\*) From gasolineFixed Where TAMEMktShare <> 0.0 and TAMEVolume = 0.0; -- ( 0)

# Select count(\*) From gasolineFixed Where TAMEMktShare = 0.0 and TAMEVolume <> 0.0; -- ( 0)

# Select count(\*) From gasolineFixed Where TAMEMktShare = 0.0 and TAMEVolume = 0.0; -- (3279)

# Select count(\*) From gasolineFixed Where TAMEMktShare <> 0.0 and TAMEVolume <> 0.0; -- (1182)

# Select count(\*) from gasolineFixed; -- (4461)

**C) NGSulfur must be an integer value equal to the NGId.**

-- Check 142.769

select NGId,

Count(\*)

From countyYearMonthFixed

Group by NGId

Order by NgId;

Update countyYearMonthFixed

set NgId = 30

where ngid =1;

select \* from naturalGas;

Select \* from naturalGasfixed;

update naturalGasFixed

Set NGId = 30;

update naturalGasFixed

Set NgSulfur = 30

Where NGId = 30;

# D) Check 89.715 -- Relative Humidity Range must be within 0 to 100.

To resolve all problems with humidity, all of the temperatures and humidity values for all calendar years, months, hours and locations were replaced with the original data submitted by Air Improvement Resources (AIR), without the subsequent updates using state supplied data.

drop table if exists countyyearmonthhour;

CREATE TABLE `countyyearmonthhour` (

`FIPSCountyId` smallint(6) NOT NULL DEFAULT '0',

`FIPSStateId` tinyint(2) NOT NULL DEFAULT '0',

`Year` smallint(6) NOT NULL DEFAULT '0',

`Month` tinyint(2) NOT NULL DEFAULT '0',

`HourID` tinyint(2) NOT NULL DEFAULT '0',

`Temperature` float DEFAULT NULL,

`RelativeHumidity` float DEFAULT NULL,

PRIMARY KEY (`FIPSCountyId`,`FIPSStateId`,`Year`,`Month`,`HourID`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1 ;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

1999 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour1999

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2000 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2000

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2001 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2001

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2002 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2002

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2003 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2003

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2004 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2004

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2005 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2005

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2006 as year,

month,

hourid,

averagetemp,

averagerelhumidity

from countymonthhour2006

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2007 as year,

month,

hourid,

averagetemp,

averagerelhumidity

from countymonthhour2007

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

fipscountyid,

2008 as year,

month,

hourid,

averagetemp,

averagerelhumidity

from countymonthhour2008

;

-- Add Broomfield County, Colorado (using Boulder County).

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

14 as fipscountyid,

2002 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2002

where

fipsstateid=8

and fipscountyid=13

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

14 as fipscountyid,

2003 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2003

where

fipsstateid=8

and fipscountyid=13

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

14 as fipscountyid,

2004 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2004

where

fipsstateid=8

and fipscountyid=13

;

insert into countyyearmonthhour (

fipsstateid,

fipscountyid,

year,

month,

hourid,

temperature,

relativehumidity )

select

fipsstateid,

14 as fipscountyid,

2005 as year,

month,

hourid,

temperature,

relativehumidity

from countymonthhour2005

where

fipsstateid=8

and fipscountyid=13

;

# Change Log for NCD20090531

The changes from the previous database, NCD20090430, are re-interpolation of fuels, elimination of working tables, and resetting CountyMonthHour to the twenty-year averages.

# Procedure

1. Copy NCD20090430 and rename NCD20090531
2. Copy gasoline and countyyearmonth tables from fuelinterp\_20090609 into NCD20090531, replacing existing files. These are the product of the fuels reinterpolation for 2006-2011 described below.
3. Delete all other countyyearmonth and gasoline tables. These tables are working tables and don't belong in a distributed database.
4. Testing countyyearmonthhour and countymonthhour: extensive testing compiled into NCDTesting\_NCD20090531.xls, including comparisons with ncd20080727 and ncd20070912.
   1. countymonthhourncd agrees in all three tables
   2. countymonthhour is wrong—same as 2008; presumably left over from Pechan's
5. Copy countymonthhourNCD into countymonthhour
6. Delete countymonthhour2005 and countymonthhour2007
7. Leave countymonthhourNCD. This table is valuable, because so often we need to modify countymonthhour and want to be able to replace it with the original.
8. Delete countyvmt1999 and countyvmt2002. These were used for VMT weighting, but can be recreated, and probably are best done for gas only. At any rate, they are not part of the standard distribution.

# Fuels Interpolation Procedure for 2006-2011

1. Interpolate all fuel properties between 2005 and 2012 at the county-month level, except BenzeneContent, and all fuel sulfur content, including GasSulfur, GasMaxSulfur, HwyDieselid,, NRDieselid, and RMDieselID. These non-interpolated parameters should be left as they were in NCD20090327 for the year and month of interest. The final benzene modification is to set 2011=2012 (see below). The 2005 fuels were developed from gas station survey data and PADD refinery data by Dave Brzezinski. The 2012 fuels were those used in the RFS2 NPRM based on AEO 2007 and refinery modeling.
2. Interpolation basis
   1. For 2005, use database NCD20090327 tables countyyearmonth and gasoline.
   2. For 2012 use database rfs2frm2022fuels, tables countyyearmonthrfs2frmfuels2022aeo and gasolinerfs2frmfuels2022aeo. Fuels are assumed the same in 2012 as in 2022.
   3. Databases are available as NCD20090327.zip and rfs2frm2022fuels.zip at H:\AMD\MOBILE\NMIM\CountyDB.
   4. Tables suitable for interpolation are created by joining countyyearmonth and gasoline, where hwygasolineid=gasolineid.
   5. The governing principle for oxygenates is to conserve them in a reasonable way. Hence we are allowing tiny market shares if the procedure produces it and are allowing EtOH vol% to rise to 10.6 if necessary to prevent the sum of market shares from exceeing unity.
3. Make no adjustments to 2005 EtOH. Our general approach has been to set all EtOH vol% to 10. However, 2005 values were estimated from survey data. In the cases where we spread remaining ethanol around to unsurveyed counties, it might have made more sense to set EtOH vol% to 10, but it is not clear exactly where we should do this, since some survey data is outside RFG areas.
4. Non-Interpolated parameters
   1. Benzene
      1. For 2006-2010, retain benzene values for those years from NCD20090327. This benzene is quite close to 2005, but comes down a little in 2010, as would be expected as refiners move toward the meeting the standard in 2011. Rationale from Rich Cook: benzene is expensive to remove and refiners won't do it if they don't have to.
      2. For 2011, set benzene to 2012 benzene. Rationale: 2012 includes the MSAT benzene controls, which begin in 2011.
   2. Sulfur—GasSulfur, GasMaxSulfur, HwyDieselid,, NRDieselid, and RMDieselID.

All should retain the values for 2006-2011 from NCD20090327. These sulfur values were previously developed to reflect the gasoline and diesel sulfur regulations and expected phase-ins.

1. Interpolation
   1. MTBE, ETBE, and TAME
      1. Interpolate market share and retain the 2005 volume %.
      2. Beginning in April 2007
         1. In RFG areas, convert these oxygenates to EtOH (see method below), and then zero them out.
         2. In non-RFG areas, simply zero them out; do not convert to EtOH.
   2. Oxygenates through March of 2007
      1. For EtOH
         1. Interpolate vms = vol%\*mktshr.
         2. Calculate EtohMktShare, ms = vms/10, thus assuming that all EtOH is E10.
         3. If the sum of ms for all oxygenates <= 1, you are done
         4. If the sum of ms for all oxygenates >1,
            1. Diminish EtOH ms so that the sum of ms for all oxygenates = 1.
            2. Calculate EtOH v% = EtOH vms / EtOH ms
            3. A QA test is that EtOH v% <= 10.6.
   3. Oxygenates beginning in April of 2007
      1. In RFG areas: Convert non-EtOH oxygenates in RFG areas to E10 beginning April of 2007:
         1. Convert all four oxygenates from vol% to wt% by multiplying vol% by the following conversion factors:
            1. EtOH: 0.3488
            2. MTBE: 0.1786
            3. ETBE: 0.1533
            4. TAME: 0.1636
         2. For all four oxygenates, calculate wtms = wt%\*market share
         3. Sum all four wtms and divide by 0.3488 to get an equivalent vol%\*market share for EtOH. That is, EtOH vms = ∑wtms / 0.3488.
         4. Divide the result of the last calculation by 10 to get an equivalent market share of E10. That is EtOH ms = EtOH vms / 10.
            1. If ms <= 1, set EtOHVol%=10.
            2. If ms > 1

Set ms = 1

Set EtOHVolume% = vms

The test for this procedure is that EtOH v% <= 10.6.

* + - 1. Set the volumes and market shares of the other three oxygenates to zero.
    1. In non-RFG areas:
       1. Set non-EtOH oxygenates in non-RFG areas to zero 2007.
       2. Calculate the EtOH mkt share by assuming EtOHvol%=10
       3. If ms > 1, adjust the EtOH v% upward so that the mkt share =1.
    2. Rationale for these oxygenate procedures beginning in April of 2007: Converting oxygenates in all areas to E10 results in a discontinuity in ethanol "volumes" (our surrogate for volume: vol%\*mktshare). Lester Wyborny explained that RFG areas require ethanol in order to replace other oxygenates, which resulted in a shift of ethanol to RFG areas, meaning at least some non-RFG areas lose ethanol. The discontinuity (at least on a national level) disappeared when we converted non-ethanol oxygenates to E10 only in RFG areas.
  1. Interpolate all other fuel properties between 2005 and 2012 at the county-month level.

# QA

QA script EtohMktShare.plx showed that no EtOH volumes exceeded 10.6 and no sums of market shares exceeded 1. VMT-weighted average BenzeneContent drops to 0.6163 in 2011. VMT-weighted average Sum of oxygen weight percent and EtOH weight percent show the following pattern:



# Fuels Interpolation Script for 2006-2011

The script is long, but provides absolute documentation. The final script that Amanda Valente ran is fuelinterpscript\_20090609.sql, pasted below.

DROP DATABASE IF EXISTS fuelInterp;

CREATE DATABASE fuelInterp;

USE fuelInterp;

### Fuel Interpolation Script

### Written by Amanda Valente 5/18/2009

### Output database is FuelInterp(gasoline, countyyearmonth)

### Input Databases NCD20090327(gasoline, countyyearmonth), rfs2frm2022fuels (gasolinerfs2frmfuels2022aeo, countyyearmonthrfs2frmfuels2022aeo)

### If changing input database/tables, change them here. Interpolation is independent, as separate tables are created.

DROP TABLE IF EXISTS gasoline2005;

CREATE TABLE gasoline2005 SELECT \* FROM NCD20090327.gasoline;

CREATE UNIQUE INDEX index1 ON gasoline2005(gasolineID);

DROP TABLE IF EXISTS gasoline2012;

CREATE TABLE gasoline2012 SELECT \* FROM rfs2frm2022fuels.gasolinerfs2frmfuels2022aeo;

CREATE UNIQUE INDEX index2 ON gasoline2012(gasolineID);

DROP TABLE IF EXISTS countyYearMonth2005;

CREATE TABLE countyYearMonth2005 SELECT \* FROM NCD20090327.countyyearmonth WHERE year BETWEEN 1999 AND 2011;

CREATE INDEX index3 ON countyYearMonth2005(hwygasolineID);

DROP TABLE IF EXISTS countyYearMonth2012;

CREATE TABLE countyYearMonth2012 SELECT \* FROM rfs2frm2022fuels.countyyearmonthrfs2frmfuels2022aeo;

CREATE INDEX index4 ON countyYearMonth2012(hwygasolineID);

### End of input database/table use.

### Creates an index for each county/month for ease in joining.

DROP TABLE IF EXISTS fuels19992011;

CREATE TABLE fuels19992011 SELECT \* , FIPSStateID\*1000000+FIPSCountyID\*1000+Month as CountyMonthID

FROM gasoline2005 INNER JOIN countyyearmonth2005 WHERE gasoline2005.gasolineID=countyyearmonth2005.hwygasolineID;

CREATE INDEX index5 ON fuels19992011(CountyMonthID);

DROP TABLE IF EXISTS sulfur;

CREATE TABLE sulfur SELECT CountyMonthID, year, GasSulfur, GasMaxSulfur, HwyDieselID, NRDieselID, RMDieselID, BenzeneContent FROM fuels19992011;

CREATE INDEX index6 ON sulfur(CountyMonthID);

DROP TABLE IF EXISTS fuels2005;

CREATE TABLE fuels2005 SELECT \* FROM fuels19992011 WHERE year=2005;

CREATE INDEX index7 ON fuels2005(CountyMonthID);

#UPDATE fuels2005 SET EtOHMktShare=EtOHVolume\*EtOHMktShare/10;

#UPDATE fuels2005 SET EtOHVolume=10 WHERE EtOHMktShare>0;

DROP TABLE IF EXISTS fuels2012;

CREATE TABLE fuels2012 SELECT \*, FIPSStateID\*1000000+FIPSCountyID\*1000+Month as CountyMonthID

FROM gasoline2012 INNER JOIN countyyearmonth2012 WHERE gasoline2012.gasolineID=countyyearmonth2012.hwygasolineID;

CREATE INDEX index8 ON fuels2012(CountyMonthID);

### The remainder of this script interpolates the values for fuel properties between 2005 and 2012.

### A separate table is created for each year, and then they are merged.

DROP TABLE IF EXISTS fuels2006;

CREATE TABLE fuels2006

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2006)/(2005-2012) AS RVP,

sulfur.GasSulfur,

sulfur.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-(fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2005-2006)/(2005-2012) AS VMS,

fuels2005.MTBEVolume,

fuels2005.MTBEMktShare-(fuels2005.MTBEMktShare-fuels2012.MTBEMktShare)\*(2005-2006)/(2005-2012) AS MTBEMktShare,

fuels2005.ETBEVolume,

fuels2005.ETBEMktShare-(fuels2005.ETBEMktShare-fuels2012.ETBEMktShare)\*(2005-2006)/(2005-2012) AS ETBEMktShare,

fuels2005.TAMEVolume,

fuels2005.TAMEMktShare-(fuels2005.TAMEMktShare-fuels2012.TAMEMktShare)\*(2005-2006)/(2005-2012) AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2006)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2006)/(2005-2012) AS OlefinContent,

sulfur.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2006)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2006)/(2005-2012) AS E300,

fuels2005.RFG,

fuels2005.GasolineID,

fuels2005.RVPOxyWaiver,

fuels2005.FIPSCountyID,

fuels2005.FIPSStateID,

2006 as Year,

fuels2005.month,

fuels2005.NGId,

sulfur.HwyDieselID,

fuels2005.HwyGasolineID,

fuels2005.NRGasolineID,

sulfur.NRDieselID,

sulfur.RMDieselID,

fuels2005.HwyFuelDataSource,

fuels2005.NRFuelDataSource,

fuels2005.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN sulfur WHERE sulfur.year=2006 AND sulfur.CountyMonthID=fuels2005.CountyMonthID AND sulfur.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2012.CountyMonthID;

UPDATE fuels2006 SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2006 SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2006 SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2006 SET EtOHMktShare=1 WHERE EtOHMktShare>1;

DROP TABLE IF EXISTS fuels2007;

CREATE TABLE fuels2007

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2007)/(2005-2012) AS RVP,

sulfur.GasSulfur,

sulfur.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-(fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2005-2007)/(2005-2012) AS vms,

fuels2005.MTBEVolume,

fuels2005.MTBEMktShare-(fuels2005.MTBEMktShare-fuels2012.MTBEMktShare)\*(2005-2007)/(2005-2012) AS MTBEMktShare,

fuels2005.ETBEVolume,

fuels2005.ETBEMktShare-(fuels2005.ETBEMktShare-fuels2012.ETBEMktShare)\*(2005-2007)/(2005-2012) AS ETBEMktShare,

fuels2005.TAMEVolume,

fuels2005.TAMEMktShare-(fuels2005.TAMEMktShare-fuels2012.TAMEMktShare)\*(2005-2007)/(2005-2012) AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2007)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2007)/(2005-2012) AS OlefinContent,

sulfur.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2007)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2007)/(2005-2012) AS E300,

fuels2005.RFG,

fuels2005.GasolineID,

fuels2005.RVPOxyWaiver,

fuels2005.FIPSCountyID,

fuels2005.FIPSStateID,

2007 as Year,

fuels2005.month,

fuels2005.NGId,

sulfur.HwyDieselID,

fuels2005.HwyGasolineID,

fuels2005.NRGasolineID,

sulfur.NRDieselID,

sulfur.RMDieselID,

fuels2005.HwyFuelDataSource,

fuels2005.NRFuelDataSource,

fuels2005.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN sulfur WHERE sulfur.year=2007 AND sulfur.CountyMonthID=fuels2005.CountyMonthID AND sulfur.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.month<4;

UPDATE fuels2007 SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2007 SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2007 SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2007 SET EtOHMktShare=1 WHERE EtOHMktShare>1;

DROP TABLE IF EXISTS fuels2007apr;

CREATE TABLE fuels2007apr

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2007)/(2005-2012) AS RVP,

sulfur.GasSulfur,

sulfur.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-(fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2005-2007)/(2005-2012) AS vms,

(fuels2005.EtOHVolume\*0.3488\*fuels2005.EtOHMktShare+fuels2005.MTBEVolume\*0.1786\*fuels2005.MTBEMktShare+fuels2005.ETBEVolume\*0.1533\*fuels2005.ETBEMktShare+fuels2005.TAMEVolume\*0.1636\*fuels2005.TAMEMktShare)/0.3488-((fuels2005.EtOHVolume\*0.3488\*fuels2005.EtOHMktShare+fuels2005.MTBEVolume\*0.1786\*fuels2005.MTBEMktShare+fuels2005.ETBEVolume\*0.1533\*fuels2005.ETBEMktShare+fuels2005.TAMEVolume\*0.1636\*fuels2005.TAMEMktShare)-(fuels2012.EtOHVolume\*0.3488\*fuels2012.EtOHMktShare+fuels2012.MTBEVolume\*0.1786\*fuels2012.MTBEMktShare+fuels2012.ETBEVolume\*0.1533\*fuels2012.ETBEMktShare+fuels2012.TAMEVolume\*0.1636\*fuels2012.TAMEMktShare))\*(2005-2007)/(0.3488\*(2005-2012)) as vmsRFG,

fuels2005.MTBEVolume,

fuels2005.MTBEMktShare-(fuels2005.MTBEMktShare-fuels2012.MTBEMktShare)\*(2005-2007)/(2005-2012) AS MTBEMktShare,

fuels2005.ETBEVolume,

fuels2005.ETBEMktShare-(fuels2005.ETBEMktShare-fuels2012.ETBEMktShare)\*(2005-2007)/(2005-2012) AS ETBEMktShare,

fuels2005.TAMEVolume,

fuels2005.TAMEMktShare-(fuels2005.TAMEMktShare-fuels2012.TAMEMktShare)\*(2005-2007)/(2005-2012) AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2007)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2007)/(2005-2012) AS OlefinContent,

sulfur.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2007)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2007)/(2005-2012) AS E300,

fuels2005.RFG,

fuels2005.GasolineID,

fuels2005.RVPOxyWaiver,

fuels2005.FIPSCountyID,

fuels2005.FIPSStateID,

2007 as Year,

fuels2005.month,

fuels2005.NGId,

sulfur.HwyDieselID,

fuels2005.HwyGasolineID,

fuels2005.NRGasolineID,

sulfur.NRDieselID,

sulfur.RMDieselID,

fuels2005.HwyFuelDataSource,

fuels2005.NRFuelDataSource,

fuels2005.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN sulfur WHERE sulfur.year=2007 AND sulfur.CountyMonthID=fuels2005.CountyMonthID AND sulfur.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.month>=4;

### NON-ETHANOL OXYGENATES ARE REMOVED FROM ALL GASOLINE BY APRIL, 2007

### Convert oxygenate vol% to wt% using

### ETOH\*0.3488

### MTBE\*0.1786

### ETBE\*0.1533

### TAME\*0.1636

UPDATE fuels2007apr SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2007apr SET EtOHVolume=10 WHERE vmsRFG>0 AND RFG='Y';

UPDATE fuels2007apr SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2007apr SET EtOHMktShare=vmsRFG/EtOHVolume WHERE EtOHVolume>0 AND RFG='Y';

UPDATE fuels2007apr SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2007apr SET EtOHVolume=vmsRFG WHERE EtOHMktShare>1 AND RFG='Y';

UPDATE fuels2007apr SET EtOHMktShare=1 WHERE EtOHMktShare>1;

UPDATE fuels2007apr SET MTBEMktShare=0;

UPDATE fuels2007apr SET TAMEMktShare=0;

UPDATE fuels2007apr SET ETBEMktShare=0;

UPDATE fuels2007apr SET MTBEVolume=0;

UPDATE fuels2007apr SET TAMEVolume=0;

UPDATE fuels2007apr SET ETBEVolume=0;

ALTER TABLE fuels2007apr DROP COLUMN vmsRFG;

### Note: all oxygenate content is interpolated linearly between 2008 and 2012, due to MTBE Removal

### Remaining parameters are interpolated linearly between 2005 and 2012

DROP TABLE IF EXISTS fuels2008;

CREATE TABLE fuels2008

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2008)/(2005-2012) AS RVP,

sulfur.GasSulfur,

sulfur.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-(fuels2005.EtOHMktShare\*fuels2005.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2005-2008)/(2005-2012) AS vms,

(fuels2005.EtOHVolume\*0.3488\*fuels2005.EtOHMktShare+fuels2005.MTBEVolume\*0.1786\*fuels2005.MTBEMktShare+fuels2005.ETBEVolume\*0.1533\*fuels2005.ETBEMktShare+fuels2005.TAMEVolume\*0.1636\*fuels2005.TAMEMktShare)/0.3488-((fuels2005.EtOHVolume\*0.3488\*fuels2005.EtOHMktShare+fuels2005.MTBEVolume\*0.1786\*fuels2005.MTBEMktShare+fuels2005.ETBEVolume\*0.1533\*fuels2005.ETBEMktShare+fuels2005.TAMEVolume\*0.1636\*fuels2005.TAMEMktShare)-(fuels2012.EtOHVolume\*0.3488\*fuels2012.EtOHMktShare+fuels2012.MTBEVolume\*0.1786\*fuels2012.MTBEMktShare+fuels2012.ETBEVolume\*0.1533\*fuels2012.ETBEMktShare+fuels2012.TAMEVolume\*0.1636\*fuels2012.TAMEMktShare))\*(2005-2008)/(0.3488\*(2005-2012)) as vmsRFG,

fuels2005.MTBEVolume,

fuels2005.MTBEMktShare-(fuels2005.MTBEMktShare-fuels2012.MTBEMktShare)\*(2005-2008)/(2005-2012) AS MTBEMktShare,

fuels2005.ETBEVolume,

fuels2005.ETBEMktShare-(fuels2005.ETBEMktShare-fuels2012.ETBEMktShare)\*(2005-2008)/(2005-2012) AS ETBEMktShare,

fuels2005.TAMEVolume,

fuels2005.TAMEMktShare-(fuels2005.TAMEMktShare-fuels2012.TAMEMktShare)\*(2005-2008)/(2005-2012) AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2008)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2008)/(2005-2012) AS OlefinContent,

sulfur.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2008)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2008)/(2005-2012) AS E300,

fuels2005.RFG,

fuels2005.GasolineID,

fuels2005.RVPOxyWaiver,

fuels2005.FIPSCountyID,

fuels2005.FIPSStateID,

2008 as Year,

fuels2005.month,

fuels2005.NGId,

sulfur.HwyDieselID,

fuels2005.HwyGasolineID,

fuels2005.NRGasolineID,

sulfur.NRDieselID,

sulfur.RMDieselID,

fuels2005.HwyFuelDataSource,

fuels2005.NRFuelDataSource,

fuels2005.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN sulfur WHERE sulfur.year=2008 AND sulfur.CountyMonthID=fuels2005.CountyMonthID AND sulfur.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2012.CountyMonthID;

CREATE INDEX index7 ON fuels2008(CountyMonthID);

UPDATE fuels2008 SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2008 SET EtOHVolume=10 WHERE vmsRFG>0 AND RFG='Y';

UPDATE fuels2008 SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2008 SET EtOHMktShare=vmsRFG/EtOHVolume WHERE EtOHVolume>0 AND RFG='Y';

UPDATE fuels2008 SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2008 SET EtOHVolume=vmsRFG WHERE EtOHMktShare>1 AND RFG='Y';

UPDATE fuels2008 SET EtOHMktShare=1 WHERE EtOHMktShare>1;

UPDATE fuels2008 SET MTBEMktShare=0;

UPDATE fuels2008 SET TAMEMktShare=0;

UPDATE fuels2008 SET ETBEMktShare=0;

UPDATE fuels2008 SET MTBEVolume=0;

UPDATE fuels2008 SET TAMEVolume=0;

UPDATE fuels2008 SET ETBEVolume=0;

ALTER TABLE fuels2008 DROP COLUMN vmsRFG;

DROP TABLE IF EXISTS fuels2009;

CREATE TABLE fuels2009

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2009)/(2005-2012) AS RVP,

sulfur.GasSulfur,

sulfur.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-(fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2008-2009)/(2008-2012) AS vms,

0 as MTBEVolume,

0 AS MTBEMktShare,

0 as ETBEVolume,

0 AS ETBEMktShare,

0 as TAMEVolume,

0 AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2009)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2009)/(2005-2012) AS OlefinContent,

sulfur.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2009)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2009)/(2005-2012) AS E300,

fuels2005.RFG,

fuels2005.GasolineID,

fuels2005.RVPOxyWaiver,

fuels2005.FIPSCountyID,

fuels2005.FIPSStateID,

2009 as Year,

fuels2005.month,

fuels2005.NGId,

sulfur.HwyDieselID,

fuels2005.HwyGasolineID,

fuels2005.NRGasolineID,

sulfur.NRDieselID,

sulfur.RMDieselID,

fuels2005.HwyFuelDataSource,

fuels2005.NRFuelDataSource,

fuels2005.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN fuels2008 INNER JOIN sulfur WHERE sulfur.year=2009 AND sulfur.CountyMonthID=fuels2005.CountyMonthID AND sulfur.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2008.CountyMonthID AND fuels2008.CountyMonthID=fuels2012.CountyMonthID;

UPDATE fuels2009 SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2009 SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2009 SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2009 SET EtOHMktShare=1 WHERE EtOHMktShare>1;

DROP TABLE IF EXISTS fuels2010;

CREATE TABLE fuels2010

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2010)/(2005-2012) AS RVP,

sulfur.GasSulfur,

sulfur.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-(fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2008-2010)/(2008-2012) AS vms,

0 as MTBEVolume,

0 AS MTBEMktShare,

0 as ETBEVolume,

0 AS ETBEMktShare,

0 as TAMEVolume,

0 AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2010)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2010)/(2005-2012) AS OlefinContent,

sulfur.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2010)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2010)/(2005-2012) AS E300,

fuels2005.RFG,

fuels2005.GasolineID,

fuels2005.RVPOxyWaiver,

fuels2005.FIPSCountyID,

fuels2005.FIPSStateID,

2010 as Year,

fuels2005.month,

fuels2005.NGId,

sulfur.HwyDieselID,

fuels2005.HwyGasolineID,

fuels2005.NRGasolineID,

sulfur.NRDieselID,

sulfur.RMDieselID,

fuels2005.HwyFuelDataSource,

fuels2005.NRFuelDataSource,

fuels2005.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN fuels2008 INNER JOIN sulfur WHERE sulfur.year=2010 AND sulfur.CountyMonthID=fuels2005.CountyMonthID AND sulfur.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2008.CountyMonthID AND fuels2008.CountyMonthID=fuels2012.CountyMonthID;

UPDATE fuels2010 SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2010 SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2010 SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2010 SET EtOHMktShare=1 WHERE EtOHMktShare>1;

### Due to MSAT Requirements, benzene levels are set to 2012 levels in 2011 since Benzene must be removed in these years.

DROP TABLE IF EXISTS fuels2011;

CREATE TABLE fuels2011

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2011)/(2005-2012) AS RVP,

sulfur.GasSulfur,

sulfur.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-(fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2008-2011)/(2008-2012) AS vms,

0 as MTBEVolume,

0 AS MTBEMktShare,

0 as ETBEVolume,

0 AS ETBEMktShare,

0 as TAMEVolume,

0 AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2011)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2011)/(2005-2012) AS OlefinContent,

fuels2012.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2011)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2011)/(2005-2012) AS E300,

fuels2005.RFG,

fuels2005.GasolineID,

fuels2005.RVPOxyWaiver,

fuels2005.FIPSCountyID,

fuels2005.FIPSStateID,

2011 as Year,

fuels2005.month,

fuels2005.NGId,

sulfur.HwyDieselID,

fuels2005.HwyGasolineID,

fuels2005.NRGasolineID,

sulfur.NRDieselID,

sulfur.RMDieselID,

fuels2005.HwyFuelDataSource,

fuels2005.NRFuelDataSource,

fuels2005.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN fuels2008 INNER JOIN sulfur WHERE sulfur.year=2011 AND sulfur.CountyMonthID=fuels2005.CountyMonthID AND sulfur.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2008.CountyMonthID AND fuels2008.CountyMonthID=fuels2012.CountyMonthID;

UPDATE fuels2011 SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2011 SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2011 SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2011 SET EtOHMktShare=1 WHERE EtOHMktShare>1;

### MUST CREATE NEW 2012 SO COLUMNS ARE IN THE RIGHT ORDER (NOTE countyyearmonthrfs2frmfuels2022aeo are in different order)

DROP TABLE IF EXISTS fuels2012B;

CREATE TABLE fuels2012B

SELECT fuels2005.RVP-(fuels2005.RVP-fuels2012.RVP)\*(2005-2012)/(2005-2012) AS RVP,

fuels2012.GasSulfur,

fuels2012.GasMaxSulfur,

fuels2005.EtOHVolume,

fuels2005.EtOHMktShare,

fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-(fuels2008.EtOHMktShare\*fuels2008.EtOHVolume-fuels2012.EtOHMktShare\*fuels2012.EtOHVolume)\*(2008-2012)/(2008-2012) AS vms,

0 as MTBEVolume,

0 AS MTBEMktShare,

0 as ETBEVolume,

0 AS ETBEMktShare,

0 as TAMEVolume,

0 AS TAMEMktShare,

fuels2005.AromaticContent-(fuels2005.AromaticContent-fuels2012.AromaticContent)\*(2005-2012)/(2005-2012) AS AromaticContent,

fuels2005.OlefinContent-(fuels2005.OlefinContent-fuels2012.OlefinContent)\*(2005-2012)/(2005-2012) AS OlefinContent,

fuels2012.BenzeneContent,

fuels2005.E200-(fuels2005.E200-fuels2012.E200)\*(2005-2012)/(2005-2012) AS E200,

fuels2005.E300-(fuels2005.E300-fuels2012.E300)\*(2005-2012)/(2005-2012) AS E300,

fuels2012.RFG,

fuels2012.GasolineID,

fuels2012.RVPOxyWaiver,

fuels2012.FIPSCountyID,

fuels2012.FIPSStateID,

2012 as Year,

fuels2012.month,

fuels2012.NGId,

fuels2012.HwyDieselID,

fuels2012.HwyGasolineID,

fuels2012.NRGasolineID,

fuels2012.NRDieselID,

fuels2012.RMDieselID,

fuels2012.HwyFuelDataSource,

fuels2012.NRFuelDataSource,

fuels2012.CountyMonthID

FROM fuels2005 INNER JOIN fuels2012 INNER JOIN fuels2008 WHERE fuels2005.CountyMonthID=fuels2012.CountyMonthID AND fuels2005.CountyMonthID=fuels2008.CountyMonthID AND fuels2008.CountyMonthID=fuels2012.CountyMonthID;

UPDATE fuels2012B SET EtOHVolume=10 WHERE vms>0;

UPDATE fuels2012B SET EtOHMktShare=vms/EtOHVolume WHERE EtOHVolume>0;

UPDATE fuels2012B SET EtOHVolume=vms WHERE EtOHMktShare>1;

UPDATE fuels2012B SET EtOHMktShare=1 WHERE EtOHMktShare>1;

### Combine all separate year fuel tables into one fuel table

DROP TABLE IF EXISTS temp;

CREATE TABLE temp SELECT \* FROM fuels2006;

INSERT INTO temp SELECT \* FROM fuels2007;

INSERT INTO temp SELECT \* FROM fuels2007apr;

INSERT INTO temp SELECT \* FROM fuels2008;

INSERT INTO temp SELECT \* FROM fuels2009;

INSERT INTO temp SELECT \* FROM fuels2010;

INSERT INTO temp SELECT \* FROM fuels2011;

INSERT INTO temp SELECT \* FROM fuels2012B;

ALTER TABLE temp DROP COLUMN vms;

DROP TABLE IF EXISTS fuels19992012;

CREATE TABLE fuels19992012 SELECT \* FROM fuels19992011 WHERE Year<2005;

INSERT INTO fuels19992012 SELECT \* FROM fuels2005;

INSERT INTO fuels19992012 SELECT \* FROM temp;

### Drop intermediate tables

DROP TABLE fuels2005;

DROP TABLE fuels2006;

DROP TABLE fuels2007;

DROP TABLE fuels2007apr;

DROP TABLE fuels2008;

DROP TABLE fuels2009;

DROP TABLE fuels2010;

DROP TABLE fuels2011;

DROP TABLE fuels2012;

DROP TABLE fuels2012B;

DROP TABLE gasoline2005;

DROP TABLE gasoline2012;

DROP TABLE CountyYearMonth2005;

DROP TABLE CountyYearMonth2012;

DROP TABLE sulfur;

#DROP TABLE fuels19992011;

### Separate fuels19992012 table into separate countyYearMonth and Gasoline tables.

CREATE TABLE Gasoline

SELECT DISTINCT RVP, GasSulfur,

GasMaxSulfur, RVPOxyWaiver, ETOHVolume, ETOHMktShare,

MTBEVolume, MTBEMktShare, ETBEVolume, ETBEMktShare,

TAMEVolume, TAMEMktShare, AromaticContent, OlefinContent,

BenzeneContent, E200, E300, RFG

FROM fuels19992012;

CREATE INDEX index10 ON Gasoline(RVP);

CREATE INDEX index11 ON fuels19992012(RVP);

ALTER TABLE Gasoline

ADD Gasolineid smallint(6) NOT NULL PRIMARY KEY AUTO\_INCREMENT;

UPDATE fuels19992012 as a, gasoline as b

SET a.gasolineid = b.gasolineid

WHERE

a.RVP = b.RVP and

a.GasSulfur = b.GasSulfur and

a.GasMaxSulfur = b.GasMaxSulfur and

a.RVPOxyWaiver = b.RVPOxyWaiver and

a.ETOHVolume = b.ETOHVolume and

a.ETOHMktShare = b.ETOHMktShare and

a.MTBEVolume = b.MTBEVolume and

a.MTBEMktShare = b.MTBEMktShare and

a.ETBEVolume = b.ETBEVolume and

a.ETBEMktShare = b.ETBEMktShare and

a.TAMEVolume = b.TAMEVolume and

a.TAMEMktShare = b.TAMEMktShare and

a.AromaticContent = b.AromaticContent and

a.OlefinContent = b.OlefinContent and

a.BenzeneContent = b.BenzeneContent and

a.E200= b.e200 and

a.e300 = b.e300 and

a.RFG = b.RFG ;

UPDATE fuels19992012 SET hwygasolineid=gasolineid;

UPDATE fuels19992012 SET nrgasolineid=hwygasolineid;

DROP TABLE IF EXISTS CountyYearMonth;

CREATE TABLE CountyYearMonth (

`FIPSCountyID` smallint(6) NOT NULL DEFAULT '0',

`FIPSStateID` tinyint(2) NOT NULL DEFAULT '0',

`Year` smallint(6) NOT NULL DEFAULT '0',

`Month` tinyint(2) NOT NULL DEFAULT '0',

`NGId` tinyint(4) DEFAULT NULL,

`HwyDieselID` smallint(6) DEFAULT NULL,

`HwyGasolineID` smallint(6) DEFAULT NULL,

`NRGasolineID` smallint(6) DEFAULT NULL,

`NRDieselID` smallint(6) DEFAULT NULL,

`RMDieselID` smallint(6) DEFAULT NULL,

`HwyFuelDataSource` smallint(6) DEFAULT NULL,

`NRFuelDataSource` smallint(6) DEFAULT NULL,

PRIMARY KEY (FIPSCountyID, FIPSStateID, Year, Month)

) ;

INSERT INTO CountyYearMonth (FIPSCountyID, FIPSStateID, Year, Month, NGId, HwyDieselID, HwyGasolineID, NRGasolineID, NRDieselID, RMDieselID, HwyFuelDataSource, NRFuelDataSource)

SELECT FIPSCountyID,

FIPSStateID,

Year,

Month,

NGId,

HwyDieselID,

HwyGasolineID,

NRGasolineID,

NRDieselID,

RMDieselID,

HwyFuelDataSource,

NRFuelDataSource

FROM fuels19992012;

### Copy over 2012 as all future years to 2050

DROP TABLE IF EXISTS temp;

CREATE TABLE temp SELECT \* FROM CountyYearMonth WHERE Year=2012;

UPDATE temp SET Year=2013;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2014;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2015;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2016;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2017;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2018;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2019;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2020;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2021;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2022;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2023;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2024;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2025;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2026;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2027;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2028;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2029;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2030;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2031;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2032;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2033;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2034;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2035;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2036;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2037;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2038;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2039;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2040;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2041;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2042;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2043;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2044;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2045;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2046;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2047;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2048;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2049;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

UPDATE temp SET Year=2050;

INSERT INTO CountyYearMonth SELECT \* FROM temp;

#DROP TABLE IF EXISTS fuels19992012;

DROP TABLE IF EXISTS temp;

### End of Script.

### Remaining segment can be used to test property ratios to ensure correct joining.

#DROP TABLE IF EXISTS joinCYMgasoline;

#CREATE TABLE joinCYMgasoline

# SELECT g.RVP,

# g.GasSulfur,

# g.GasMaxSulfur,

# g.EtOHVolume,

# g.EtOHMktShare,

# g.MTBEVolume,

# g.MTBEMktShare,

# g.ETBEVolume,

# g.ETBEMktShare,

# g.TAMEVolume,

# g.TAMEMktShare,

# g.AromaticContent,

# g.OlefinContent,

# g.BenzeneContent,

# g.E200,

# g.E300,

# g.RFG,

# g.GasolineID,

# g.RVPOxyWaiver,

# c.FIPSCountyID,

# c.FIPSStateID,

# c.Year,

# c.Month,

# c.NGID,

# c.HWYDieselID,

# c.HWYGasolineID,

# c.NRGasolineID,

# c.NRDieselID,

# c.RMDieselID,

# c.HwyFuelDataSource,

# c.NRFuelDataSource,

# c.FIPSStateID\*1000000+c.FIPSCountyID\*1000+c.Month as CountyMonthID

# FROM gasoline AS g INNER JOIN CountyYearMonth AS c

# WHERE g.gasolineid=c.hwygasolineid;

#CREATE INDEX index50 ON joincymgasoline(CountyMonthID);

#DROP TABLE IF EXISTS test;

#CREATE TABLE test

# SELECT j.RVP/f.RVP as RVP,

# j.GasSulfur/f.GasSulfur as GasSulfur,

# j.GasMaxSulfur/f.GasMaxSulfur as GasMaxSulfur,

# j.EtOHVolume/f.EtOHVolume as EtOHVolume,

# j.EtOHMktShare/f.EtOHMktShare as EtOHMktShare,

# j.MTBEVolume/f.MTBEVolume as MTBEVolume,

# j.MTBEMktShare/f.MTBEMktShare as MTBEMktShare,

# j.ETBEVolume/f.ETBEVolume as ETBEVolume,

# j.ETBEMktShare/f.ETBEMktShare as ETBEMktShare,

# j.TAMEVolume/f.TAMEVolume as TAMEVolume,

# j.TAMEMktShare/f.TAMEMktShare as TAMEMktShare,

# j.AromaticContent/f.AromaticContent as AromaticContent,

# j.OlefinContent/f.OlefinContent as OlefinContent,

# j.BenzeneContent/f.BenzeneContent as BenzeneContent,

# j.E200/f.E200 as E200,

# j.E300/f.E300 as E300,

# j.Countymonthid/f.countymonthid as CountyMonthID,

# j.year/f.year as year

# FROM joincymgasoline as j INNER JOIN fuels19992012 as f WHERE j.countymonthid=f.countymonthid AND j.year=f.year;

# Change Log for NCD20090430

From John Van Bruggen of Pechan. This version copies the interpolation procedure listed in NCD20090302(years 2007 and 2008) but applies to years 2006, 2009,2010, and 2011

# Change Log for NCD20090327

From John Van Bruggen of Pechan. The version corrects the interpolation done for 2007 and 2008 in two states, MD and MI. The two end point years of the interpolation had oxygenate market shares under ETOH only (ie ETOHMktShare = 1) but for years 2007 and 2008 there were also small percentages of TAME and ETBE being reported. To resolve this issue, the ETOHMktShare was updated such that

ETOHMktShare = 1-(ETBEMktShare + TAMEMktShare)

for all gasoline profiles in the states MD and MI. This resulted in changes to 80 gasoline profiles.

The other changes were the addition of 2008 temperature and Relative Humidity to the countyyearmonthhour table and replacement of countymonthhour with countymonthhour2008. Both of these tables were sent by David Brzezinski to Pechan on March 26, 2009

# ChangeLog for NCD20090302

This database contains updates to fix issues with the way gasoline profiles were incorporated for 2005, as well as updates to temperatures in 2006 and 2007, VMT in 2007 and 2008, and creation of new gasoline profiles for use in 2007 and 2008.

At the request of MD, Pechan updated gasoline IDs 286-297 with the assumption that these profiles were not being used by any other state-county-month-year combinations. While this assumption worked for 2005, these profiles were used in other years. The gasoline table was replaced with the table that originally came from NCD20070912. Next state supplied profiles were brought in from NCD20080727 using the query:

insert into NCD20090302.gasoline

select \* from NCD20080727.gasoline where gasolineid>4150

After that, gasoline profiles 286-297 from NCD20080727 were then renumbered and inserted into the gasoline table with gasoline IDs 4188-4199. CountyYearMonth was also updated for MD in 2005 to point to Gasoline IDs 4188-4199 instead of 286-297.

Next, another issue found on further review was that gasoline profiles were only updated for Nonroad in 2005 in MI and WI. HwyGasolineID was set equal to NRGasolineID in the CountyYearMonth table for 2005.

Temperature data was provided by OTAQ in the form of MySQL tables for 2006 and 2007. CountyYearMonthHour had additional records added using the following queries:

insert into NCD20090302.countyyearmonthhour

select \* from NCD20090302.countyyearmonthhour2006 where year = 2006

insert into NCD20090302.countyyearmonthhour

select \* from NCD20090302.countyyearmonthhour2007 where year = 2007

2007 VMT data was compiled by Pechan using HPMS data sources to estimate VMT. 1082928 records were added to the baseyearvmt table.

2008 VMT data was estimated using the Federal Highway Administration (FHWA) Office of Highway Policy Information’s Traffic Volume Trends (TVT) data (<http://www.fhwa.dot.gov/ohim/tvtw/tvtpage.cfm>) in estimating 2008 VMT for the NEI.  TVT provides national data in the following six road categories: Rural Interstate, Rural Other Arterial, Other Rural, Urban Interstate, Urban Other Arterial, and Other Urban.  Data are also provided at the state level for the combined Rural Interstate and Rural Other Arterial categories and the combined Urban Interstate and Urban Other Arterial categories.  The national data are reported by month for 2007 and 2008, cumulative by month for 2007 and 2008, and with the percent change from 2007 to 2008.  The state-level data are reported only for the latest month for 2007 and 2008, along with the percent change from 2007 to 2008.

1082928 records were added added to the baseyearvmt table.

Gasoline Interpolation:

The last time interpolation was done, 2009 values were determined from AEO2007 (see changes under NCD20070912). This time intertoplated values were calculated for 2007 and 2008 using 2005 and 2012.

1. Allocate 2005 gasoline values to county, year, and month using the following query:

Select FIPSStateID, FIPSCountyID, year, month, NRGasolineID, RVP, ETOHVol, ETOHMktShare,MTBEVol, MTBEMktShare

from NCD20090302.countyyearmonth cym, NCD20090302.gasoline g

where cym.NRGasolineID = g.GasolineID and cym.year = 2005

1. Use similar query to create 2012 table at county month level using NmimRfs2Fuels2022AEO developed by OTAQ
2. Convert MTBE volumes in 2005 gasolines to corresponding ETOH volumes using the following formula (Vol % ETOH= vol % MTBE \* (0.1786/0.3488)) and compute weighted average 2005 ETOH Volumes based on the original 2005 ETOH volumes, the ETOH volume to be converted from MTBE and the original 2005 market shares for ETOH and MBTE:

*Adj 2005 ETOH Vol = (2005 ETOH Vol \* 2005 ETOH MktShare + 2005 MTBE Vol \* (0.1786/0.3488) \* 2005 MTBE MktShare)/(2005 ETOH MktShare + 2005 MTBE MktShare)*

1. Calculate Adjusted 2005 ETOH Market Share by adding the 2005 ETOH and MTBE market shares :

*Adj 2005 ETOH MktShare = 2005 ETOH MktShare +2005 MTBE MktShare*

1. Replace all 2005 MBTE Vol and MTBE MktShare with 0
2. Calculate 2007 ETOH Mkt Share by interpolation using the adjusted 2005 ETOH market shares calculated in Step 3 above and the 2012 ETOH market shares:

*2007 ETOH MktShare = Adj 2005 ETOH MktShare +((2012 ETOH MktShare-Adj 2005 ETOH MktShare)\*((2007-2005)/(2012-2005)))*

1. Calculate 2007 ETOH Vol by interpolation (again using the adjusted 2005 ETOH volumes and market shares calculated in Steps 3 and 4 above):

*2007 ETOH Vol = ((Adj 2005 ETOH Vol\*Adj 2005 ETOH MktShare]) +((2012 ETOH Vol \*2012 ETOH MktShare)-(Adj 2005 ETOH Vol \* Adj 2005 ETOH MktShare))\*(2007-2005/2012-2005))/ (2005 Adj ETOH MktShare +((2012 ETOH MktShare –Adj 2005 ETOH MktShare)\*((2007-2005)/(2012-2005))))*

1. Calculate 2007 RVP by interpolation from 2005 and 2012 RVP values:

*2007 RVP = 2005\_RVP+((2012\_RVP-2005\_ RVP)\*((2007-2005)/(2012-2005)))*

1. Extract remaining 2007 gasoline parameters using existing 2007 values from the NCD with the following query:

Select FIPSStateID, FIPSCountyID, year, month, NRGasolineID, GasSulfur, GasMaxSulfur,

MTBEVolume, MTBEMktShare, ETBEVolume, ETBEMktShare, TAMEVolume, TAMEMktShare,

AromaticContent, OlefinContent, BenzeneContent, E200, E300, RFG, rvpoxywaiver

from NCD20090302.countyyearmonth cym, NCD20090302.gasoline g

where cym.NRGasolineID = g.GasolineID and cym.year = 2007

1. Create new table by combining Interpolated RVP, ETOH Vol, and ETOH MktShare with 2007 values from Step 9 on state, county, month basis.
2. Create new records to add to NCD gasoline table by performing a grouping query on previous table (from Step 10) based on 2005, 2007, and 2012 gasoline IDs
3. Add additional column [new\_gas\_id] to create new gasoline IDs for each record starting with the next integer past the previous maximum in the gasoline table
4. Update countyyearmonth table with the following query :

select fipsstateid, fipscountyid, year, month, new\_gas\_id as NRGasolineID, new\_gas\_id as HwyGasolineID

from step 10 table ten, step 11 table eleven

where ten.2005 Gas ID=eleven.2005 Gas ID and ten.2007 Gas ID=eleven.2007 Gas ID

and ten.2012 Gas ID = eleven.2012.Gas ID

Steps 6-13 were repeated for 2008 changing the year where appropriate.

# ChangeLog for NCD20080727

This database continues changes begun in NCD20080724 to fix problems with state submittals in the NEI2005v2 database from Pechan, and to project state submittals into future years, in order to start onroad runs for RFS2.

The previous database, NCD20080724, was codified as a finished database to start the RFS2 nonroad runs. The problems listed at the bottom of ChangeLog for NCD20080724 are as follows, but I believe that not all of them have to be dealt with as indicated in italics below:

Problems that remain for the next NCD version before onroad NMIM is run are

a) 91 Utah (49) speed VMT files that will have to be changed. *Tests indicate that having "SPEED VMT" as the first line or comments before the data block do not cause any problem. Hence these files are OK.*

b) look at imc files, which seem odd. *For now I am assuming that these files are OK. Dave would have to examine them more closely.*

c) Project forward again to cover these files. *Given that the two above points are now considered non-problems, forward projection is unnecessary.*

d) create all future dsf files

e) check all dsf files for date on first line. *I have now checked these. All have the date on the first line. Tested by looking for length >4 using checkLine1DSF.plx.*

So creating future DSF files are all that remain, plus adding correct 2030 VMT from ncd20080522 (not Pechan's version).

## Creating Future DSF Files

# CreateFutureYearNewDSF.plx

# Print all new DSF files to get a look at them prior to generating

# future ones.

$dir="E:\\mysql\\data\\NCD20080727\\ExternalFiles";

foreach $file (

'1100105.dsf',

'5101305.DSF',

'5105905.DSF',

'5110705.DSF',

'5115305.DSF',

'5117905.DSF',

'5151005.DSF',

'5160005.DSF',

'5161005.DSF',

'5168305.DSF',

'5168505.DSF',

'2400105.dsf',

'2400305.DSF',

'2401505.DSF',

'2402905.dsf',

'2403505.DSF',

'2404305.DSF',

) {

print "file = $file\n";

$baseyr=substr($file,5,2);

print "baseyr = $baseyr\n";

$cnty=substr($file,0,5);

print "cnty = $cnty\n";

$inpath="$dir\\$file";

print "inpath = $inpath\n";

# Loop through future year dsf files.

# for ($yr=$baseyr+1;$yr<$baseyr+2;$yr=$yr+1){ #diagnostic

for ($yr=$baseyr+1;$yr<$baseyr+25;$yr=$yr+1){

print "yr = $yr\n";

$year=2000+$yr;

print "year = $year\n";

if($yr<10){

$outfile=$cnty.'0'.$yr.'.dsf';

}else{

$outfile="$cnty$yr.dsf";

}

print "outfile = $outfile\n";

$outpath="$dir\\$outfile";

print "outpath = $outpath\n";

open (out1,">$outpath");

open (in1,"$inpath");

$lcnt=1;

# Read the baseyear file and modify for each future year file.

while (<in1>) {

# Print the date on the first line:

if($lcnt==1){print out1 "$year\n";}

# Print subsequent 14 lines:

else {

@fieldlist=split(/ /,$\_);

# Repeat the first field $yr-$baseyr+1 times.

for ($field=0;$field<$yr-$baseyr+1;$field=$field+1){

print out1 $fieldlist[0].' ';

}

# Add in subsequent fields

for ($field=1;$field<25-($yr-$baseyr);$field=$field+1){

print out1 $fieldlist[$field].' ';

}

print out1 "\n";

}

$lcnt=$lcnt+1;

}

close(in1);

close(out1);

}

}

## Updating CountyYear Table with New DieselFractFileName

# CountyYearFutureYearNewDSF.plx

# Modify CountyYear table to reference new DSF files.

$ncd='ncd20080727';

foreach $baseyearfile (

'1100105.dsf',

'5101305.DSF',

'5105905.DSF',

'5110705.DSF',

'5115305.DSF',

'5117905.DSF',

'5151005.DSF',

'5160005.DSF',

'5161005.DSF',

'5168305.DSF',

'5168505.DSF',

'2400105.dsf',

'2400305.DSF',

'2401505.DSF',

'2402905.dsf',

'2403505.DSF',

'2404305.DSF',

) {

print "baseyearfile = $baseyearfile\n";

$baseyr=substr($baseyearfile,5,2);

print "baseyr = $baseyr\n";

$baseyear=2000+$baseyr;

print "baseyear = $baseyear\n";

$state=substr($baseyearfile,0,2);

print "state = $state\n";

$cnty=substr($baseyearfile,2,3);

print "cnty = $cnty\n";

$cntycode=substr($baseyearfile,0,5);

print "cntycode = $cntycode\n";

# for ($yr=$baseyr+0;$yr<$baseyr+2;$yr=$yr+1){ # diagnostic

for ($yr=$baseyr+0;$yr<$baseyr+25;$yr=$yr+1){

print "yr = $yr\n";

$year=2000+$yr;

print "year = $year\n";

if($yr<10){

$dieselfractfilename=$cntycode.'0'.$yr;

}else{

$dieselfractfilename="$cntycode$yr";

}

print "dieselfractfilename = $dieselfractfilename\n";

$sql="

use $ncd;

update countyyear

set dieselfractfilename=$dieselfractfilename

where

fipsstateid=$state

and fipscountyid=$cnty

and year=$year

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

}

# Set all future years to the last dieselfractfilename

print "year = $year\n";

print "dieselfractfilename = $dieselfractfilename\n";

$sql="

use $ncd;

update countyyear

set dieselfractfilename=$dieselfractfilename

where

fipsstateid=$state

and fipscountyid=$cnty

and year>$year

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

}

## Adding Correct 2030 VMT

2030 BaseYearVMT table was inserted from NCD20080522 using the script that follows. This table was created by OTAQ to fix the 2030 VMT problem in PF02v3. Note the important distinction between NCD20080522 and NCD20080522Pechan2005NEIv2. Changelogs for both are included below.

# AddNew2030VMT.plx

# This script adds the corrected 2030 VMT from NCD20080522.

# I was tempted to add the whole file, but some changes could

# have been made to 2005, so I thought it best to just add 2030

$outfile='AddNew2030VMT.txt';

$ncd='NCD20080727';

$sql="

use $ncd;

# QA

select

'BeforeReplacement' as ccase,

Baseyear,

count(\*),

sum(vmt)

from baseyearvmt

group by baseyear

;

# Delete 2030

delete from baseyearvmt where baseyear=2030;

# Insert new vmt

insert into baseyearvmt (

BaseYear,

VClass,

RoadType,

FIPSCountyId,

FIPSStateId,

VMT

)

select

BaseYear,

VClass,

RoadType,

FIPSCountyId,

FIPSStateId,

VMT

from ncd20080522.baseyearvmt

where baseyear=2030

;

# QA

select

'AfterReplacement' as ccase,

Baseyear,

count(\*),

sum(vmt)

from baseyearvmt

group by baseyear

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql >> $outfile`;

As is indicated by the QA results below, only 2030 changed.

|  |  |  |  |
| --- | --- | --- | --- |
| ccase | Baseyear | count(\*) | sum(vmt) |
| BeforeReplacement | 1999 | 1082592 | 2,684,203 |
| BeforeReplacement | 2002 | 1082592 | 2,840,778 |
| BeforeReplacement | 2005 | 1082592 | 3,026,525 |
| BeforeReplacement | 2009 | 1082592 | 3,164,653 |
| BeforeReplacement | 2010 | 1082592 | 3,244,086 |
| BeforeReplacement | 2014 | 1082592 | 3,500,837 |
| BeforeReplacement | 2015 | 1082592 | 3,567,133 |
| BeforeReplacement | 2020 | 1082592 | 3,916,831 |
| BeforeReplacement | 2030 | 1082592 | 4,685,156 |
| ccase | Baseyear | count(\*) | sum(vmt) |
| AfterReplacement | 1999 | 1082592 | 2,684,203 |
| AfterReplacement | 2002 | 1082592 | 2,840,778 |
| AfterReplacement | 2005 | 1082592 | 3,026,525 |
| AfterReplacement | 2009 | 1082592 | 3,164,653 |
| AfterReplacement | 2010 | 1082592 | 3,244,086 |
| AfterReplacement | 2014 | 1082592 | 3,500,837 |
| AfterReplacement | 2015 | 1082592 | 3,567,133 |
| AfterReplacement | 2020 | 1082592 | 3,916,831 |
| AfterReplacement | 2030 | 1082592 | 4,685,228 |

## QA:

Number of external files in NCD20080727: 9336

Number of external files in NCD20080522Pechan2005NEIv2: 9138

Number of external files in NCD20070912: 8910

Number of external files in NCD20080522: 8910

# ChangeLog for NCD20080724

Copied NCD20080522Pechan2005NEIv2, which is the database that Pechan gave us as the one they used to run NEI2005v2, and which contains all state-submitted inputs, including corrections up to this date.

This database will be set up for NR first, since there are numerous onroad external file problems that still need to be solved.

## NONROAD

26000.POP. Craig looked at 26000.pop submitted by MI (modified February 15, 2006, 6:03:00 PM) and compared it to the one from ncd20070912 (modified November 27, 2006, 2:50:52 PM). Curiously, the submitted file is older, but Craig says it does have some updates. Unfortunately, it required some changes to make it work properly with NR05. Craig made those changes (he called it 26000-MI-fix.POP), and I substituted his file and renamed it.

## ONROAD

1. 2. Changed the following external files because their program enddates were less than 2050

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| date | time | ampm | size | filename | Action |
| 9/25/2007 | 3:01 | PM | 141 | 2300005.atp | Changed 20 to 50 |
| 9/25/2007 | 2:59 | PM | 126 | 2300505.atp | Changed 20 to 50 |
| 9/25/2007 | 3:20 | PM | 231 | 2300505.imp | Changed 2025 to 2050 |

3. Copied forward all files with the following script:

# copyforward.plx

$ncd='ncd20080724';

print "ncd = $ncd\n";

$outfile="copyforward$ncd.txt";

print "outfile = $outfile\n";

$sql="

use $ncd;

update countyyear a, countyyear b

set

a.atpfilename=b.atpfilename,

a.IMFileName=b.IMFileName,

a.RegDistFileName=b.RegDistFileName,

a.VMTByHourFileName=b.VMTByHourFileName,

a.MileAccumFileName=b.MileAccumFileName,

a.TripsPerDayFileName=b.TripsPerDayFileName,

a.StartDistFileName=b.StartDistFileName,

a.SoakDistFileName=b.SoakDistFileName,

a.HotSoakFileName=b.HotSoakFileName,

a.DiurnSoakActivityFileName=b.DiurnSoakActivityFileName,

a.TripLengthFileName=b.TripLengthFileName,

a.AvgSpeedDistBaseFileName=b.AvgSpeedDistBaseFileName,

a.Stage2Pct=b.Stage2Pct,

a.NRACTFileName=b.NRACTFileName

where

a.fipsstateid=b.fipsstateid

and a.fipscountyid=b.fipscountyid

and a.year>2005

and b.year=2005

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

Problems that remain for the next NCD version before onroad NMIM is run are

a) 91 Utah (49) speed VMT files that will have to be changed.

b) look at imc files, which seem odd.

c) Project forward again to cover these files

d) create all future dsf files

e) check all dsf files for date on first line.

# ChangeLog for NCD20080522Pechan2005NEIv2

(Received from Pechan. This Change Log written by Harvey Michaels).

This is Pechan's revision of NCD20080311, which was NCD20070912 with data submitted by states as part of the 2005 NEI process. The database Pechan sent us did not include a change log and was named NCD20080522. I changed the name because OTAQ created a new version of NCD20070912 with corrected 2030 VMT, which coincidentally had exactly the same name. NCD20080522Pechan2005NEIv2 corrects the error described in point 1 of the following email from Kirstin Thesing 05/29/2008 07:45 AM. I include point 2 for completeness, but we did not think it a problem:

1. In the 2005 NCD, several onroad external reference files intended for

FIPS 51013 (Arlington Cty, VA) were used in modeling onroad emissions for

FIPS 13051 (Chatham Cty, GA). These include the diesel fraction file, speed

distribution files, and trip length files. Though the impact on national

total CAP emissions is small (see second sheet of first attached file), the

impact on emissions for county 13051 is rather large (see first sheet of

attached file). The NCD contains 2 additional incorrect speed file

references for 2 other counties (FIPS 36051 and 41051), but these 2 counties

were run before the file references were added.

2. The I/M file we used for the 2005 NEI for Maine appears to be rather

different from the version used and corrected by EPA for the 2002 NEI. When

we provided these files to you back in December, we were given the green

light to use this file, but I wonder if this was overlooked. Let me know

what you think. I have attached the prior I/M file with EPA corrections and

the latest file used for the 2005 NEI runs for your reference.

State-submitted changes to this database were only applied to 2005. They were not projected forward to future years in cases where that would be appropriate (for example, age distribution, speed, diesel sales fraction).

John Van Bruggen's email 06/18/2008 07:37 PM:

I have posted a new version of the NCD to Pechan’s FTP site

(ftp://ftp.pechan.com/public/EPA/) it is called NCD20080522.zip

The only change between the two versions is an update to the countyyear

table to fix a situation where county 51 was updated in 3 states rather

than 3 counties in state 51 (VA). I have attached the updated mysql files

(change .sip back to .zip) if you would prefer to not download a 200M

file.

John Van Bruggen

E.H. Pechan & Associates

(704) 323-7720

[attachment "countyyear.sip" deleted by Harvey Michaels/AA/USEPA/US]

# ChangeLog for NCD20080522

This database is identical to NCD20070912 except for the 2030 VMT, which has been corrected. The previous 2030 VMT was correct at the national level but slightly misallocated to states and strongly misallocated to counties.

Pechan posted a database that is the final one for the 2005 NEI v2 that is also called NCD20080522, but I have renamed it NCD20080522Pechan2005NEI.

This NCD was created by copying NCD20070912 and replacing the baseyearvmt table with baseyearvmtnew5 from E:\mysql\data\new2030vmt. Baseyearvmtnew5 was created from baseyearvmt2030new4 and ncd20070912.baseyearvmt by E:\a\NMIMRuns\PF02v3\VMT\Pechan\New2030VMT\AEOmap.sql. Both the old baseyearvmt and baseyearvmtnew5 have 9,743,328 records. The year by year totals agree, as expected. The county trend lines that previously showed anomalous behavior were rechecked and looked good.

Some useful tables not part of the formal NCD are included. Cntycntsncd1999 and cntycntsncd2002 are the number of counties by state for 1999 and 2002, which are useful for NMIM run diagnostics. Countymonthhour2002 and Countymonthhour2005 contain meteorological data from those two years, which are useful for running future years (by copying into countymonthhour) with the met data for those two years. CountymonthhourNCD is the normal countymonthhour table, containing 20-year average temperatures. Part5vclass describes the part5 vehicle classes, which correspond to those used in SCCs. M5VClass has been added to the SCC table to facilitate comparisons to the CONCEPT emissions model.

# ChangeLog for NCD20070912

# Overview

The creation of the NCD20070912 database is a repeat of the creation of the NCD20070727 database to fix problems identified in values in that version in the interpolated calendar years. None of the non-interpolated years had these problems and the NCD2007027 values are identical to the values in this updated database. The interpolation problems were:

1) All oxygenate volume values (etohvolume, mtbevolume, etbevolume and tamevolume) in interpolated years (2004, 2006, 2007, 2008, 2010 and 2011) were incorrectly calculated (zero). These values are now correctly calculated.

2) The MTBE volume and market share values (mtbevolume and mtbemktshare) for 2006 were not interpolated between calendar year 2005 and 2007 values. These values are now correctly calculated.

3) The benzene values (benzenecontent) for calendar year 2010 were originally interpolated between benzene values for calendar years 2009 and 2012. Now the 2010 values are interpolated between the benzene values in 2009 and 2011, after 2011 values are set equal to the benzene values in 2012.

The goal is a comprehensive set of fuels for NMIM based on the RFS work which was included in PF02v3. The latter included fuels for 2002, 2004, 2009, and 2012. 2012 and following years are assumed to be the same. 2004 was RFS Ref case for 2012, 2015, and 2020. Hence, some parameters (e.g., sulfur) are reflective of out years rather than 2004. Dave Brzezinski developed fuels for 2005 using a methodology for combining fuel surveys and refinery data.

All tables from 1999 to 2012 must exist in a form joined to counties, referred to as "cymgas," in order to generate a single gasoline table. (This table lacks information such as NGID, datasource, and NRdieselid, RMDieselId.) Then a unique gasoline table is extracted and gasolineids assigned. Then gasolineid is transferred to cymgas by joining it to gasoline based on gasoline properties. Then a full cymPF02v3 table can be populated by matching gasolines and updating hwygasolineid and nrgasolineid.

The resulting final files, CountyYearMonthPF02v3 and GasolinePF02v3, were copied into the CountyYearMonth and Gasoline tables of a copy of the NCD20070727 database named NCD20070912.

Many of the steps below replicate the steps taken to create the original NCD20070727 version and are necessary to generate a new set of CountyYearMonth and Gasoline tables for the new database.

# Starting Point

2002 fuels were produced by correcting RVP only per Rick Rykowski.

2005 fuels were produced for the 2005 NEI by Dave Brzezinski using historical data.

2009 fuels were produced using Brent Lignell's spreadsheets based on AEO2007.

2012 fuels were produced using Brent Lignell's spreadsheets based on AEO2007.

2013 – 2050 were set the same as 2012 per Rick Rykowski.

# Create two tables

**Script**: CopyCYMGas2007123.plx

This script copies ncd20070123.countyyearmonth and gasoline into four tables in db FuelsPF02v3:

1. countyyearmonthncd20070123

2. gasolinencd20070123

3. countyyearmonthpf02v3

4. gasolinepf02v3

The corresponding countyyearmonth and gasoline tables are joined to produce two tables:

1) cymgasncd20070123 and

2) cymgaspf02v3

These two tables are identical to start with. Cymgaspf02v3 is updated as the following series of procedures is performed. Cymgasncd20070123 remains the same and is used for comparison.

After all updates are made, unique gasolines are extracted from cymgaspf02v3 as gasolinepf02v3 and the gasolines are numbered by setting the variable gasolineid as autoincrement. These two tables are joined on gasoline properties, and gasolineid is set in cymgaspf02v3. Finally, cymgaspf02v3 is joined to CountyYearMonthpf02v3 and hwygasolineid and nrgasolineid set to gasolineid. The result is CountyYearMonthPf02v3 and GasolinePf02v3 that include all the fuels for all counties, years, and months.

# 1999, 2000, 2001, 2002, 2003

Leave as is, except for copying RVP from non-RFG areas from 2002. 2002 was prepared with Rick's help to fix the excessive RVP in non-surveyed areas that were used by ERG. However, we don't want to copy all 2002 RVP, because the RFG program changed between 1999 and 2002. Since all RFG areas are surveyed, it is assumed that those data are correct.

Source for 2002: NCD20061227pf02v3.CountyYearMonthRFS2002 and GasolineRFS2002

**Script**: UpdateRVP9903.plx

## QA

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Old | old | old | old | new | new | new | new |  |
| Tbl | year | rfg | sum(rvp) | tbl | year | rfg | sum(rvp) | new-old |
| cymgasncd20070123 | 1999 | N | 363,506 | cymgaspf02v3 | 1999 | N | 375,101 | 11,596 |
| cymgasncd20070123 | 1999 | Y | 24,892 | cymgaspf02v3 | 1999 | Y | 24,892 | 0 |
| cymgasncd20070123 | 2000 | N | 363,426 | cymgaspf02v3 | 2000 | N | 375,101 | 11,675 |
| cymgasncd20070123 | 2000 | Y | 23,994 | cymgaspf02v3 | 2000 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2001 | N | 363,553 | cymgaspf02v3 | 2001 | N | 375,224 | 11,671 |
| cymgasncd20070123 | 2001 | Y | 23,994 | cymgaspf02v3 | 2001 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2002 | N | 363,553 | cymgaspf02v3 | 2002 | N | 375,224 | 11,671 |
| cymgasncd20070123 | 2002 | Y | 23,994 | cymgaspf02v3 | 2002 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2003 | N | 363,439 | cymgaspf02v3 | 2003 | N | 375,224 | 11,785 |
| cymgasncd20070123 | 2003 | Y | 23,994 | cymgaspf02v3 | 2003 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2004 | N | 363,461 | cymgaspf02v3 | 2004 | N | 363,461 | 0 |
| cymgasncd20070123 | 2004 | Y | 23,479 | cymgaspf02v3 | 2004 | Y | 23,479 | 0 |

We observe that there is no change for RFG=Y. Sum of new RVP is higher for 2001-2003 than for 1999-2000, because there is an additional county in the countyyearmonth table beginning in 2001.

# 2005

Substitute Dave Brzezinski's 2005 table for 2005.

Source: Fuels2005.CountyYearMonth2005 and Gasoline2005

**Script**: Update2005.plx

# 2009

Substitute Rick Rykowski's 2009 table for 2009.

Source: Fuels2009.CountyYearMonthRFS2009Fix and GasolineRFS2009Fix

**Script**: Update2009.plx

# 2004

Interpolate all parameters between 2003 and 2005.

2004 is the year Tier 2 takes effect and should reflect some fuel transitioning. Since 2005 was carefully prepared by Dave Brzezinski, we feel 2004 is best prepared as an interpolation between 2003 and 2005 for all parameters.

**Script**: InterpolateFuelsAllMonths.plx. Joins cymgaspf02v3 to itself twice and interpolates.

# 2006, 2007, 2008

1. Interpolate all parameters between 2005 and 2009

**Script**: InterpolateFuelsAllMonths.plx

1. Then set 2007 and 2008 MTBE to zero and then obtain 2006 MTBE by interpolating between 2005 and 2007.

Script: Set20072008MTBEZero.plx and Interpolate2006MTBE.plx

# 2012-2050

Substitute Rick Rykowski's 2020 fuels with MSAT benzene controls for all years 2012-2050

Source: Fuels2020.CountyYearMonthRfsMsat2020 and Gasoline RfsMsat2020

**Script**: Update20122050.plx

# 2010, 2011

Interpolate all parameters between 2009 and 2012

2009 was prepared with Rick's help reflecting a good future estimate, and 2012 is the future-year EIA estimate for the RFS, which Rick said to apply to all future years.

**Script**: InterpolateFuelsAllMonths.plx

# 2011

Since MSAT takes effect in 2011, set benzene to be the same as 2012, for which it was already properly processed and then obtain 2010 benzene content by interpolating between 2009 and 2011.

**Script**: Set2011BenzTo2012.plx and Interpolate2010Benzene.plx

# Quality Assurance Checks

A check was made to make sure that there are no cases where a oxygenate volume percent is greater than zero where the oxygenate market share is zero or where the oxygenate volume percent is zero and the oxygenate market share is greater than zero. No cases were found.

**Script**: GasolineMSVolTest.plx

An example case was examined to assure that the interpolations and other adjustments were properly applied (at least in this case).

select

fipsstateid as SID,

fipscountyid as CID,

year as YR,

month as Mth,

ETOHVolume as EV,

ETOHMktShare as EM,

MTBEVolume as MV,

MTBEMktShare as MM,

BenzeneContent as BC

from cymgaspf02v3

where month=1 and

year<2013 and

fipsstateid=32 and

fipscountyid=31;

flush tables;

+-----+-----+------+-----+---------+--------+---------+--------+-------+

| SID | CID | YR | Mth | EV | EM | MV | MM | BC |

+-----+-----+------+-----+---------+--------+---------+--------+-------+

| 32 | 31 | 1999 | 1 | 0 | 0 | 4.9 | 1 | 0.5 |

| 32 | 31 | 2000 | 1 | 0 | 0 | 4.9 | 1 | 0.5 |

| 32 | 31 | 2001 | 1 | 0 | 0 | 4.9 | 1 | 0.5 |

| 32 | 31 | 2002 | 1 | 0 | 0 | 4.9 | 1 | 0.5 |

| 32 | 31 | 2003 | 1 | 0 | 0 | 4.9 | 1 | 0.5 |

| 32 | 31 | 2004 | 1 | 7.7408 | 0.495 | 4.94746 | 0.5044 | 1.1 |

| 32 | 31 | 2005 | 1 | 7.7408 | 0.99 | 10.3412 | 0.0088 | 1.7 |

| 32 | 31 | 2006 | 1 | 8.30987 | 0.9925 | 10.3412 | 0.0044 | 1.4 |

| 32 | 31 | 2007 | 1 | 8.87608 | 0.995 | 0 | 0 | 1.1 |

| 32 | 31 | 2008 | 1 | 9.43945 | 0.9975 | 0 | 0 | 0.8 |

| 32 | 31 | 2009 | 1 | 10 | 1 | 0 | 0 | 0.5 |

| 32 | 31 | 2010 | 1 | 10 | 1 | 0 | 0 | 0.495 |

| 32 | 31 | 2011 | 1 | 10 | 1 | 0 | 0 | 0.49 |

| 32 | 31 | 2012 | 1 | 10 | 1 | 0 | 0 | 0.49 |

+-----+-----+------+-----+---------+--------+---------+--------+-------+

14 rows in set (0.00 sec)

Gasoline sulfur levels should be going down as the Tier 2 fuel regulations are phased in. A check was made to see if any sulfur levels increased from calendar year 2003 to 2004.

select

a.fipsstateid as SID,

a.fipscountyid as CID,

a.year as YR,

a.month as Mth,

a.GasSulfur as AGS,

b.GasSulfur as BGS,

(b.gassulfur-a.gassulfur) as DGS

from cymgaspf02v3 as a,cymgaspf02v3 as b

where a.year=2003

and b.year=a.year+1

and a.month=b.month

and a.fipsstateid=b.fipsstateid

and a.fipscountyid=b.fipscountyid

and a.GasSulfur<b.GasSulfur

group by (b.gassulfur-a.gassulfur)

order by (b.gassulfur-a.gassulfur) desc

limit 20;

+-----+-----+------+-----+-------+---------+------------------+

| SID | CID | YR | Mth | AGS | BGS | DGS |

+-----+-----+------+-----+-------+---------+------------------+

| 29 | 47 | 2003 | 1 | 183.7 | 232.8 | 49.0999908447266 |

| 32 | 9 | 2003 | 1 | 0 | 39.985 | 39.9850006103516 |

| 6 | 1 | 2003 | 6 | 17.3 | 48.635 | 31.3350028991699 |

| 6 | 1 | 2003 | 5 | 18.03 | 49 | 30.9699993133545 |

| 6 | 1 | 2003 | 1 | 19 | 49.485 | 30.4850006103516 |

| 15 | 1 | 2003 | 1 | 24 | 51.985 | 27.9850006103516 |

| 32 | 9 | 2003 | 2 | 24.89 | 52.43 | 27.5400009155273 |

| 32 | 1 | 2003 | 1 | 30 | 54.985 | 24.9850006103516 |

| 2 | 13 | 2003 | 5 | 153 | 176.17 | 23.1699981689453 |

| 32 | 1 | 2003 | 2 | 37.11 | 58.54 | 21.4300003051758 |

| 32 | 1 | 2003 | 3 | 42.44 | 61.205 | 18.7650032043457 |

| 15 | 1 | 2003 | 5 | 43.3 | 61.635 | 18.3350028991699 |

| 32 | 9 | 2003 | 3 | 43.56 | 61.765 | 18.2049980163574 |

| 4 | 3 | 2003 | 1 | 44.6 | 62.285 | 17.685001373291 |

| 32 | 1 | 2003 | 5 | 46 | 62.985 | 16.9850006103516 |

| 4 | 3 | 2003 | 2 | 49.76 | 64.865 | 15.1049995422363 |

| 29 | 47 | 2003 | 2 | 213.4 | 226.845 | 13.4450073242188 |

| 4 | 3 | 2003 | 3 | 53.62 | 66.795 | 13.1749992370605 |

| 32 | 9 | 2003 | 5 | 56 | 67.985 | 11.9850006103516 |

| 4 | 3 | 2003 | 5 | 56.2 | 68.085 | 11.8849983215332 |

+-----+-----+------+-----+-------+---------+------------------+

20 rows in set (0.70 sec)

The minimum sulfur level allowed in MOBILE6 is 30 ppm. Using this as a limit, only three gasoline formulations had changes greater than 30 ppm. In two cases, the initial (2003) level was already below the minimum and in the third case the change was less than a 30% increase (49/184) in gasoline sulfur. All of these changes are likely within the uncertainty in the gasoline sulfur levels in a county. Although it is alarming that counties in Nevada report gasoline sulfur levels of zero (unlikely), NMIM automatically increases any value less than 30 ppm to 30 ppm.

An example case was examined to show how sulfur levels changed with calendar year. The Nevada case was selected.

select

fipsstateid as SID,

fipscountyid as CID,

year as YR,

month as Mth,

GasSulfur as GS

from cymgaspf02v3

where month=1 and

year<2014 and

fipsstateid=32 and

fipscountyid=9;

flush tables;

+-----+-----+------+-----+---------+

| SID | CID | YR | Mth | GS |

+-----+-----+------+-----+---------+

| 32 | 9 | 1999 | 1 | 0 |

| 32 | 9 | 2000 | 1 | 0 |

| 32 | 9 | 2001 | 1 | 0 |

| 32 | 9 | 2002 | 1 | 0 |

| 32 | 9 | 2003 | 1 | 0 |

| 32 | 9 | 2004 | 1 | 39.985 |

| 32 | 9 | 2005 | 1 | 79.97 |

| 32 | 9 | 2006 | 1 | 67.4775 |

| 32 | 9 | 2007 | 1 | 54.985 |

| 32 | 9 | 2008 | 1 | 42.4925 |

| 32 | 9 | 2009 | 1 | 30 |

| 32 | 9 | 2010 | 1 | 30 |

| 32 | 9 | 2011 | 1 | 30 |

| 32 | 9 | 2012 | 1 | 30 |

| 32 | 9 | 2013 | 1 | 30 |

+-----+-----+------+-----+---------+

15 rows in set (0.00 sec)

The gasoline sulfur values in 2005 are based on survey analysis and values in 2009 are based on sulfur content regulations. 2004, 2006, 2007 and 2008 are interpolations. Although the sulfur levels look chaotic, the variation is not very great. This is not a typical case, but one in which the levels were identified as anomalous. Most areas show a downward trend.

To demonstrate that no changes were made to the non-interpolated years, a comparison was made of the gasoline values in calendar year 2012 in both the NCD20070727 and NCD20070912 versions using a script. The results were identical.

**Script**: Compare2012.plx

# Separate CymGasPf02v3 into CountyYearMonthPv02v3 and GasolinePf02v3

**Script**: FuelsToNmimAllYears.plx

All of the scripts and non-standard databases referenced in this section are available on the EPA network shared drive at:

H:\AMD\MOBILE\NMIM\CountyDB\NCD20070912 Scripts

# ChangeLog for NCD20070727

External files were copied from NCD20070123. The special files that were included to enable PF02v3 runs include fuels, baseyearVMTNew3, and VMT for 2005 from OAQPS. These are now included in this database as standard files. Met data for 2003 and 2004 was updated to duplicate county 08013 into 08014. Referenced scripts are included at the end of the ChangeLog for NCD20070727.

## BaseYearVMT

Script: BaseYearVmt.plx

1. NCD20070123Pf02v3.BaseYearVmtNew3 was copied to db NCD20070727.
2. Delete 2050.
3. Insert 2005 "real" data.

## CountyYearMonthHour

This table provides met data. The biggest problem is the absence of 08014.

Script: CountyYearMonthHour2004a. Copy CountyYearMonthHour2004 into CountyYearMonthHour2004a and insert 08014 to match 08013.

Script: CountyYearMonthHour.plx

1. Copy CountyYearMonthHour from ncd20070123pf02v3 to ncd20070727.
2. Insert CountyMonthHour2005a into CountyYearMonthHour. CountyYearMonthHour2005a already has 08014.
3. Insert CountyMonthHour2004a into CountyYearMonthHour. CountyYearMonthHour2004a already has 08014.

Script: CountyYearMonthHourFix2003.plx. Adds 08014 to 2003

1. Make a separate table of 2003 08013 called cymh2003c08013
2. Insert cymh2003c08013 into CountyYearMonthHour as 08014 for 2003

## CountyMonthHour

Tests below indicate that CountyMonthHourNCD is the same as CountyMonthHour from NCD20050714, and that they both include 08014.

mysql> use ncd20050714

Database changed

mysql> select count(\*) from countymonthhour where fipsstateid=8 and fipscountyid

=13;

+----------+

| count(\*) |

+----------+

| 288 |

+----------+

1 row in set (0.03 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhour where fipsstateid=8

and fipscountyid=13;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 13509.599985123 | 288 |

+------------------+----------+

1 row in set (0.22 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhour where fipsstateid=8

and fipscountyid=14;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 13509.599985123 | 288 |

+------------------+----------+

1 row in set (0.03 sec)

mysql> use ncd20070123pf02v3

Database changed

mysql> select sum(averagetemp),count(\*) from countymonthhourncd where fipsstatei

d=8 and fipscountyid=14;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 13509.599985123 | 288 |

+------------------+----------+

1 row in set (0.01 sec)

mysql> select year,sum(temperature),count(\*) from countyyearmonthhour where fips

stateid=8 and fipscountyid=14 group by year;

+------+------------------+----------+

| year | sum(temperature) | count(\*) |

+------+------------------+----------+

| 2002 | 13603.099982262 | 288 |

+------+------------------+----------+

1 row in set (0.03 sec)

mysql> select year,sum(temperature),count(\*) from countyyearmonthhour group by y

ear;

+------+------------------+----------+

| year | sum(temperature) | count(\*) |

+------+------------------+----------+

| 1999 | 51453647.1978746 | 927936 |

| 2000 | 50511547.1989206 | 927936 |

| 2001 | 51098821.298136 | 927936 |

| 2002 | 51006061.5986269 | 928224 |

| 2003 | 50309876.597992 | 927936 |

+------+------------------+----------+

5 rows in set (2.67 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhour;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 51058503.198778 | 927936 |

+------------------+----------+

1 row in set (0.24 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhour where fipsstateid=8

and fipscountyid=14;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| NULL | 0 |

+------------------+----------+

1 row in set (0.00 sec)

mysql> select database();

+-------------------+

| database() |

+-------------------+

| ncd20070123pf02v3 |

+-------------------+

1 row in set (0.02 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhourncd where fipsstatei

d=8 and fipscountyid=14;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 13509.599985123 | 288 |

+------------------+----------+

1 row in set (0.00 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhourncd where fipsstatei

d=8 and fipscountyid=14;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 13509.599985123 | 288 |

+------------------+----------+

1 row in set (0.00 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhourncd;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 50037443.196349 | 928224 |

+------------------+----------+

1 row in set (0.28 sec)

mysql> use ncd20050714

Database changed

mysql> select sum(averagetemp),count(\*) from countymonthhour;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 50037443.196349 | 928224 |

+------------------+----------+

1 row in set (0.38 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhour where fipsstateid=8

and fipscountyid=14;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 13509.599985123 | 288 |

+------------------+----------+

1 row in set (0.02 sec)

mysql> select sum(averagetemp),count(\*) from countymonthhour where fipsstateid=8

and fipscountyid=13;

+------------------+----------+

| sum(averagetemp) | count(\*) |

+------------------+----------+

| 13509.599985123 | 288 |

+------------------+----------+

1 row in set (0.00 sec)

## Copy the rest of the tables and external files

Copy tables from NCD20070123Pf02v3 into NCD20070727, except for baseyearvmt (see above) and countyyearmonthhour (see above).

### Script: NCD20070727.plx

# ncd20070727.plx

# Script to tables from ncd20070123pf02v3 to ncd20070727

# NCD20070727 has already been supplied with

# baseyearvmt and countyyearmonthhour

$outfile='NCD20070727.txt';

open (out1,">$outfile");

$ncd='ncd20070727';

print out1 "ncd = $ncd\n";

print out1 "outfile = $outfile\n";

$todatadir="E:\\mysql\\data\\$ncd";

print out1 "todatadir = $todatadir\n";

print out1 "\n";

print out1 "fromdatadir fromtable totable\n";

foreach $dbfromtabletotable (

'NCD20070123pf02v3 county county',

'NCD20070123pf02v3 countymap countymap',

'NCD20070123pf02v3 countymonth countymonth',

'NCD20070123pf02v3 CountymonthhourNCD CountymonthhourNCD',

'NCD20070123pf02v3 CountymonthhourNCD Countymonthhour',

'NCD20070123pf02v3 Countymonthhour2005a Countymonthhour2005',

'FuelsPF02v3 Countyyearmonthpf02v3 Countyyearmonth',

'FuelsPF02v3 Gasolinepf02v3 Gasoline',

'NCD20070123pf02v3 countynrfile countynrfile',

'NCD20070123pf02v3 countyvmt1999 countyvmt1999',

'NCD20070123pf02v3 countyvmt2002 countyvmt2002',

'NCD20070123pf02v3 countyvmtmonthallocation countyvmtmonthallocation',

'NCD20070123pf02v3 countyyear countyyear',

'NCD20070123pf02v3 datasource datasource',

'NCD20070123pf02v3 diesel diesel',

'NCD20070123pf02v3 emissiontype emissiontype',

'NCD20070123pf02v3 filetype filetype',

'NCD20070123pf02v3 hour hour',

'NCD20070123pf02v3 hpmsroadtype hpmsroadtype',

'NCD20070123pf02v3 m6vclass m6vclass',

'NCD20070123pf02v3 m6vtype m6vtype',

'NCD20070123pf02v3 naturalgas naturalgas',

'NCD20070123pf02v3 part5vclass part5vclass',

'NCD20070123pf02v3 pollutantcode pollutantcode',

'NCD20070123pf02v3 scc scc',

'NCD20070123pf02v3 scctoxics scctoxics',

'NCD20070123pf02v3 state state',

'NCD20070123pf02v3 vmtmonthallocation vmtmonthallocation',

){

($db,$fromtable,$totable)=split(/ /,$dbfromtabletotable);

$fromdatadir="E:\\mysql\\data\\$db";

print out1 "$fromdatadir $fromtable $totable\n";

foreach $fileext ('frm','MYD','MYI') {

`copy $fromdatadir\\$fromtable.$fileext $todatadir\\$totable.$fileext`;

}

}

Output from this script:

|  |  |  |  |
| --- | --- | --- | --- |
| NCD = NCD20070727 |  |  |  |
| outfile = NCD20070727.txt |  |  |  |
| todatadir = E:\mysql\data\ncd20070727 |  |  |  |
|  |  |  |  |
| **fromdatadir** | **fromtable** | **totable** | **standard set** |
| E:\mysql\data\NCD20070123pf02v3 | county | county | 1 |
| E:\mysql\data\NCD20070123pf02v3 | countymap | countymap | 1 |
| E:\mysql\data\NCD20070123pf02v3 | countymonth | countymonth | 1 |
| E:\mysql\data\NCD20070123pf02v3 | Countymonthhour2005a | Countymonthhour2005 | 0 |
| E:\mysql\data\NCD20070123pf02v3 | CountymonthhourNCD | CountymonthhourNCD | 0 |
| E:\mysql\data\NCD20070123pf02v3 | CountymonthhourNCD | Countymonthhour | 1 |
| E:\mysql\data\NCD20070123pf02v3 | countynrfile | countynrfile | 1 |
| E:\mysql\data\NCD20070123pf02v3 | countyvmt1999 | countyvmt1999 | 0 |
| E:\mysql\data\NCD20070123pf02v3 | countyvmt2002 | countyvmt2002 | 0 |
| E:\mysql\data\NCD20070123pf02v3 | countyvmtmonthallocation | countyvmtmonthallocation | 1 |
| E:\mysql\data\NCD20070123pf02v3 | countyyear | countyyear | 1 |
| E:\mysql\data\FuelsPF02v3 | Countyyearmonthpf02v3 | Countyyearmonth | 1 |
| E:\mysql\data\NCD20070123pf02v3 | datasource | datasource | 1 |
| E:\mysql\data\NCD20070123pf02v3 | diesel | diesel | 1 |
| E:\mysql\data\NCD20070123pf02v3 | emissiontype | emissiontype | 1 |
| E:\mysql\data\NCD20070123pf02v3 | filetype | filetype | 1 |
| E:\mysql\data\FuelsPF02v3 | Gasolinepf02v3 | Gasoline | 1 |
| E:\mysql\data\NCD20070123pf02v3 | hour | hour | 1 |
| E:\mysql\data\NCD20070123pf02v3 | hpmsroadtype | hpmsroadtype | 1 |
| E:\mysql\data\NCD20070123pf02v3 | m6vclass | m6vclass | 1 |
| E:\mysql\data\NCD20070123pf02v3 | m6vtype | m6vtype | 1 |
| E:\mysql\data\NCD20070123pf02v3 | naturalgas | naturalgas | 1 |
| E:\mysql\data\NCD20070123pf02v3 | part5vclass | part5vclass | 1 |
| E:\mysql\data\NCD20070123pf02v3 | pollutantcode | pollutantcode | 1 |
| E:\mysql\data\NCD20070123pf02v3 | SCC | scc | 1 |
| E:\mysql\data\NCD20070123pf02v3 | scctoxics | scctoxics | 1 |
| E:\mysql\data\NCD20070123pf02v3 | state | state | 1 |
| E:\mysql\data\NCD20070123pf02v3 | vmtmonthallocation | vmtmonthallocation | 1 |
|  |  |  |  |
| already in ncd20070727 |  | baseyearvmt | 1 |
| already in ncd20070727 |  | countyyearmonthhour | 1 |
|  |  |  |  |
|  |  | sum | 26 |

## Fuels: development of all years

### Overview

The goal is a comprehensive set of fuels for NMIM based on the RFS work that was included in PF02v3. The latter included fuels for 2002, 2004, 2009, and 2012. 2012 and following years are assumed to be the same. 2004 was RFS Ref case for 2012, 2015, and 2020. Hence, some parameters (e.g., sulfur) are reflective of out years rather than 2004. Dave Brzezinski developed fuels for 2005 using a methodology for combining fuel surveys and refinery data.

All tables from 1999 to 2012 must exist in a form joined to counties, referred to as "cymgas," in order to generate a single gasoline table. (This table lacks information such as NGID, datasource, and NRdieselid, RMDieselId.) Then a unique gasoline table is extracted and gasolineids assigned. Then gasolineid is transferred to cymgas by joining it to gasoline based on gasoline properties. Then a full cymPF02v3 table can be populated by matching gasolines and updating hwygasolineid and nrgasolineid.

### Starting Point

2002 fuels were produced by correcting RVP only per Rick Rykowski.

2005 fuels were produced for the 2005 NEI by Dave Brzezinski using historical data.

2009 fuels were produced using Brent Lignell's spreadsheets based on AEO2007.

2012 fuels were produced using Brent Lignell's spreadsheets based on AEO2007. For how ethanol production was allocated to counties, the the RFS RIA.

2013 – 2050 were set the same as 2012 per Rick Rykowski.

### Create six tables

Script: CopyCYMGas2007123.plx

copies ncd20070123.countyyearmonth and gasoline into six tables in db FuelsPF02v3:

1. countyyearmonthncd20070123

2. gasolinencd20070123

3. countyyearmonthpf02v3

4. gasolinepf02v3

The corresponding countyyearmonth and gasoline tables are joined to produce two tables:

5. cymgasncd20070123 and

6. cymgaspf02v3

These ncd20070123 and pf02v3 tables are identical to start with. Cymgaspf02v3 is updated as the following series of procedures is performed. Cymgasncd20070123 remains the same and is used for comparison.

After all updates are made, unique gasolines are extracted from cymgaspf02v3 as gasolinepf02v3, and the gasolines are numbered by setting the variable gasolineid as autoincrement. These two tables are joined on gasoline properties, and gasolineid is set in cymgaspf02v3. Finally, cymgaspf02v3 is joined to CountyYearMonthpf02v3 and hwygasolineid and nrgasolineid set to gasolineid. The result is CountyYearMonthPf02v3 and GasolinePf02v3 that include all the fuels for all counties, years, and months. These final two tables were copied from database FuelsPf02v3 to NCD20070727 as CountyYearMonth and Gasoline (see below).

### 1999, 2000, 2001, 2002, 2003

These fuels were left as is, except for copying RVP from non-RFG areas from 2002. 2002 was prepared with Rick's help to fix the excessive RVP in non-surveyed areas that were used by ERG. However, we don't want to copy all 2002 RVP, because the RFG program changed between 1999 and 2002. Since all RFG areas are surveyed, it is assumed that those data are correct.

Source for 2002: NCD20061227pf02v3.CountyYearMonthRFS2002 and GasolineRFS2002

Script: UpdateRVP9903.plx

## QA

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Old | old | old | old | new | new | new | new |  |
| Tbl | year | RFG | sum(rvp) | tbl | year | rfg | sum(rvp) | new-old |
| cymgasncd20070123 | 1999 | N | 363,506 | cymgaspf02v3 | 1999 | N | 375,101 | 11,596 |
| cymgasncd20070123 | 1999 | Y | 24,892 | cymgaspf02v3 | 1999 | Y | 24,892 | 0 |
| cymgasncd20070123 | 2000 | N | 363,426 | cymgaspf02v3 | 2000 | N | 375,101 | 11,675 |
| cymgasncd20070123 | 2000 | Y | 23,994 | cymgaspf02v3 | 2000 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2001 | N | 363,553 | cymgaspf02v3 | 2001 | N | 375,224 | 11,671 |
| cymgasncd20070123 | 2001 | Y | 23,994 | cymgaspf02v3 | 2001 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2002 | N | 363,553 | cymgaspf02v3 | 2002 | N | 375,224 | 11,671 |
| cymgasncd20070123 | 2002 | Y | 23,994 | cymgaspf02v3 | 2002 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2003 | N | 363,439 | cymgaspf02v3 | 2003 | N | 375,224 | 11,785 |
| cymgasncd20070123 | 2003 | Y | 23,994 | cymgaspf02v3 | 2003 | Y | 23,994 | 0 |
| cymgasncd20070123 | 2004 | N | 363,461 | cymgaspf02v3 | 2004 | N | 363,461 | 0 |
| cymgasncd20070123 | 2004 | Y | 23,479 | cymgaspf02v3 | 2004 | Y | 23,479 | 0 |

We observe that there is no change for RFG=Y. Sum of new RVP is higher for 2001-2003 than for 1999-2000, because there is an additional county in the countyyearmonth table beginning in 2001.

### 2005

Substitute Dave Brzezinski's 2005 table for 2005.

Source: Fuels2005.CountyYearMonth2005 and Gasoline2005

Script: Update2005.plx

### 2009

Substitute Rick Rykowski's 2009 table for 2009.

Source: Fuels2009.CountyYearMonthRFS2009Fix and GasolineRFS2009Fix

Script: Update2009.plx

### 2004

Interpolate all parameters between 2003 and 2005.

2004 is the year Tier 2 takes effect and should reflect some fuel transitioning. Since 2005 was carefully prepared by Dave Brzezinski, we feel 2004 is best prepared as an interpolation between 2003 and 2005 for all parameters.

Script: InterpolateFuelsAllMonths.plx. Joins cymgaspf02v3 to itself twice and interpolates.

### 2006, 2007, 2008

1. Interpolate all parameters between 2005 and 2009

Script: InterpolateFuelsAllMonths.plx

1. Then set 2007 and 2008 MTBE to zero, and then obtain 2006 MTBE by interpolating between 2005 and 2007.

Script: Set20072008MTBEZero.plx

### 2012-2050

Substitute Rick Rykowski's 2020 fuels with MSAT benzene controls for all years 2012-2050

Source: Fuels2020.CountyYearMonthRfsMsat2020 and Gasoline RfsMsat2020

Script: Update20122050.plx

### 2010, 2011

Interpolate all parameters between 2009 and 2012

2009 was prepared with Rick's help reflecting a good future estimate, and 2012 is the future-year EIA estimate for the RFS, which Rick said to apply to all future years.

Script: InterpolateFuelsAllMonths.plx

### 2011

Since MSAT takes effect in 2011, set benzene to be the same as 2012, for which it was already properly processed.

Script: Set2011BenzTo2012.plx

### Separate CymGasPf02v3 into CountyYearMonthPv02v3 and GasolinePf02v3

Script: FuelsToNmimAllYears.plx

## Scripts referenced above

### Script: BaseYearVmt.plx

# BaseYearVmt.plx

# Script to compile all Vmt table for ncd20070123pf02v3

# for inclusion into ncd20070727

$srcncd='ncd20070123pf02v3';

print "srcncd = $srcncd\n";

$ncd='ncd20070727';

print "ncd = $ncd\n";

$outfile='BaseYearVmt.txt';

print "outfile = $outfile\n";

# Copy baseyearvmtnew3

`mysql -e "flush tables;"`;

$fromdatadir="E:\\mysql\\data\\$srcncd";

print "fromdatadir = $fromdatadir\n";

$todatadir="E:\\mysql\\data\\$ncd";

print "todatadir = $todatadir\n";

$fromtable="baseyearvmtnew3";

print "fromtable = $fromtable\n";

$totable="baseyearvmt";

print "totable = $totable\n";

foreach $fileext ('frm','MYD','MYI') {

print "fileext = $fileext\n";

`copy $fromdatadir\\$fromtable.$fileext $todatadir\\$totable.$fileext`;

}

$sql="

use $ncd;

## Delete 2050

delete from baseyearvmt where baseyear=2050

;

select baseyear,count(\*)

from baseyearvmt

group by baseyear

order by baseyear

;

# Include 2005

insert baseyearvmt (

BaseYear,

VClass,

RoadType,

FipsStateId,

FipsCountyId,

VMT

)

select

BaseYear,

VClass,

RoadType,

FipsStateId,

FipsCountyId,

VMT

from $srcncd.baseyearvmt2005

order by

BaseYear,

VClass,

RoadType,

FipsStateId,

FipsCountyId

;

# Diagnostic

select

baseyear,

count(\*),

sum(vmt)

from baseyearvmt

group by baseyear

order by baseyear

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql >> $outfile`;

### Script: CountyYearMonthHour2004a

# CountyMonthHour2004a.plx

# Script to add Broomfield County CO, 08014, to countmonthhour2004

# in met data for ncd20070123pf02v3

# for inclusion into ncd20070727

# It is a copy of Boulder County, 08013

$ncd='ncd20070123pf02v3';

print "ncd = $ncd\n";

$outfile='CountyMonthHour2004a.txt';

print "outfile = $outfile\n";

# Copy countymonthhour2004 into countymonthhour2004a.

`mysql -e "flush tables;"`;

$fromdatadir="E:\\mysql\\data\\$ncd";

print "fromdatadir = $fromdatadir\n";

$todatadir="E:\\mysql\\data\\$ncd";

print "todatadir = $todatadir\n";

$fromtable="countymonthhour2004";

print "fromtable = $fromtable\n";

$totable="countymonthhour2004a";

print "totable = $totable\n";

foreach $fileext ('frm','MYD','MYI') {

print "fileext = $fileext\n";

`copy $fromdatadir\\$fromtable.$fileext $todatadir\\$totable.$fileext`;

}

$sql="

use $ncd;

insert countymonthhour2004a (

FipsStateId,

FipsCountyId,

Month,

HourId,

Temperature,

RelativeHumidity

)

select

8 as FipsStateId,

14 as FipsCountyId,

Month,

HourId,

Temperature,

RelativeHumidity

from countymonthhour2004

where

fipsstateid=8

and fipscountyid=13

order by

FipsStateId,

FipsCountyId,

Month,

HourId

;

# Diagnostic

select \* from countymonthhour2004a

where fipsstateid=8 and fipscountyid in(13,14)

order by month,hourid

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql > $outfile`;

### Script: CountyYearMonthHour.plx

# CountyYearMonthHour.plx

# Script to compile met data for ncd20070123pf02v3

# for inclusion into ncd20070727

# Inserts countymonthhour2004a and 2005a into countyyearmonthhour

$srcncd='ncd20070123pf02v3';

print "srcncd = $srcncd\n";

$ncd='ncd20070727';

print "ncd = $ncd\n";

$outfile='CountyYearMonthHour.txt';

print "outfile = $outfile\n";

# Copy countyyearmonthhour. Has 1999,2000,01,02,03

`mysql -e "flush tables;"`;

$fromdatadir="E:\\mysql\\data\\$srcncd";

print "fromdatadir = $fromdatadir\n";

$todatadir="E:\\mysql\\data\\$ncd";

print "todatadir = $todatadir\n";

$fromtable="countyyearmonthhour";

print "fromtable = $fromtable\n";

$totable="countyyearmonthhour";

print "totable = $totable\n";

foreach $fileext ('frm','MYD','MYI') {

print "fileext = $fileext\n";

`copy $fromdatadir\\$fromtable.$fileext $todatadir\\$totable.$fileext`;

}

$sql="

use $ncd;

# 2005a

insert countyyearmonthhour (

Year,

FipsStateId,

FipsCountyId,

Month,

HourId,

Temperature,

RelativeHumidity

)

select

2005 as Year,

FipsStateId,

FipsCountyId,

Month,

HourId,

AverageTemp,

AverageRelHumidity

from $srcncd.countymonthhour2005a

order by

Year,

FipsStateId,

FipsCountyId,

Month,

HourId

;

# 2004a

insert countyyearmonthhour (

Year,

FipsStateId,

FipsCountyId,

Month,

HourId,

Temperature,

RelativeHumidity

)

select

2004 as Year,

FipsStateId,

FipsCountyId,

Month,

HourId,

Temperature,

RelativeHumidity

from $srcncd.countymonthhour2004a

order by

Year,

FipsStateId,

FipsCountyId,

Month,

HourId

;

# Diagnostic

select

'$ncd' as ncd,

year,

count(\*),

avg(temperature),

avg(relativehumidity)

from countyyearmonthhour

group by year

order by year

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql > $outfile`;

### Script: CountyYearMonthHourFix2003.plx

# CountyYearMonthHourFix2003.plx

# Script to add Broomfield County, CO, 08014, to 2003

# in met data in ncd20070727.CountyYearMonthHour

$ncd='ncd20070727';

print "ncd = $ncd\n";

$outfile='CountyYearMonthHourFix2003.txt';

print "outfile = $outfile\n";

$sql="

use $ncd;

# Create a separate table of 08013 for 2003

drop table if exists cymh2003c08013;

create table cymh2003c08013

select \* from countyyearmonthhour

where

fipsstateid=8

and fipscountyid=13

and year=2003

;

# Insert the new table of 08013 for 2003 into 08014

insert countyyearmonthhour (

Year,

FipsStateId,

FipsCountyId,

Month,

HourId,

Temperature,

RelativeHumidity

)

select

2003 as Year,

8 as FipsStateId,

14 as FipsCountyId,

Month,

HourId,

Temperature,

RelativeHumidity

from cymh2003c08013

order by

FipsStateId,

FipsCountyId,

Month,

HourId

;

# Diagnostic

select \* from countyyearmonthhour

where fipsstateid=8 and fipscountyid in(13,14) and year=2003

order by month,hourid

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql > $outfile`;

### Script: CopyCYMGas2007123.plx

# CopyCYMGas20070123.plx

# Copy CYM and Gas from NCD20070123 and join them into a single

# table, which will serve as the framework for all the gasoline

# changes from RFS and interpolation

# This is the first step in generating a whole set of pf02v3 fuels

# Two sets of tables are created:

# One with the suffix 20070123 and one with the suffix pf02v3

# The first is the source and will remain unchanged;

# the second will be the final set with all changes worked in.

# Both sets are retained for QA comparison purposes

$ncd='ncd20070123';

print "ncd = $ncd\n";

$db='fuelsPF02v3';

print "db = $db\n";

$outfile='CopyCymGas20070123.txt';

$fromdatadir="E:\\mysql\\data\\$ncd";

$todatadir="E:\\mysql\\data\\$db";

print "fromdatadir = $fromdatadir\n";

print "todatadir = $todatadir\n";

# Copy Tables

foreach $table (

'Countyyearmonth',

'Gasoline',

) {

`mysql -e "flush tables;"`;

$fromtable="$table";

$totable1="${table}$ncd";

$totable2="${table}PF02v3";

print "totable1 = $totable1\n";

print "totable2 = $totable2\n";

foreach $fileext ('frm','MYD','MYI') {

print "fileext = $fileext\n";

`copy $fromdatadir\\$fromtable.$fileext $todatadir\\$totable1.$fileext`;

`copy $fromdatadir\\$fromtable.$fileext $todatadir\\$totable2.$fileext`;

}

}

# Join cym and gasoline for two tables, one for a check and the other

# for updating

# Set CA RFG=Y for both cymgas tables

foreach $version(

'pf02v3',

"$ncd",

){

print "version = $version\n";

$sql="

use $db;

drop table if exists cymgas$version;

create table cymgas$version

select

fipsstateid,

fipscountyid,

year,

month,

RVP,

GasSulfur,

GasMaxSulfur,

EtohVolume,

EtohMktShare,

MtbeVolume,

MtbeMktShare,

EtbeVolume,

EtbeMktShare,

TameVolume,

TameMktShare,

AromaticContent,

OlefinContent,

BenzeneContent,

E200,

E300,

RFG

from

countyyearmonth$version c,

gasoline$version g

where c.hwygasolineid=g.gasolineid

;

alter table cymgas$version

add primary key (fipsstateid,fipscountyid,year,month)

;

update cymgas$version

set rfg='Y' where fipsstateid=6

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

}

# Diagnostics

$sql="

use $db;

select '$ncd' as db,'$ncd.countyyearmonth' as tbl,count(\*)

from $ncd.countyyearmonth

;

select '$db' as db,'countyyearmonth$ncd' as tbl, count(\*)

from countyyearmonth$ncd

;

select '$db' as db,'countyyearmonthpf02v3' as tbl, count(\*)

from countyyearmonthpf02v3

;

select '$db' as db,'cymgas$ncd' as tbl,count(\*)

from cymgas$ncd

;

select distinct RFG,'$db' as db,'cymgas$ncd' as tbl

from cymgas$ncd

where fipsstateid=6

;

select '$db' as db,'cymgaspf02v3' as tbl,count(\*)

from cymgaspf02v3

;

select '$db' as db,'cymgaspf02v3' as tbl,rfg,count(rfg)

from cymgaspf02v3

where fipsstateid=6

group by rfg

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql > $outfile`;

### Script: UpdateRVP9903.plx

# UpdateRVP9903.plx

# Copy RVP from the RFS 2002 except for RFG counties,

# including all of CA

$db="fuelspf02v3";

print "db = $db\n";

$ncd="ncd20070123";

print "ncd = $ncd\n";

$outfile="UpdateRVP9903.txt";

print "outfile = $outfile\n";

$sql="

use $db;

update

cymgaspf02v3 c,

ncd20061227pf02v3.countyyearmonthrfs2002 d,

ncd20061227pf02v3.gasolinerfs2002 g

set c.rvp=g.rvp

where

d.year=2002

and c.year in(1999,2000,2001,2002,2003)

and c.fipsstateid=d.fipsstateid

and c.fipscountyid=d.fipscountyid

and c.month=d.month

and c.RFG='N'

and d.hwygasolineid=g.gasolineid

;

# Test

select

'cymgas$ncd' as tbl,

year,

rfg,

sum(rvp)

from cymgas$ncd

group by year,rfg

order by year,rfg

;

select

'cymgaspf02v3' as tbl,

year,

rfg,

sum(rvp)

from cymgaspf02v3

group by year,rfg

order by year,rfg

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

#`mysql < script.sql`;

`mysql < script.sql > $outfile`;

### Script: Update2005.plx

# Update2005.plx

# Substitute Dave Brzezinski's 2005 table for 2005.

# Source: Fuels2005.CountyYearMonth2005 and Gasoline2005

$db="fuelspf02v3";

print "db = $db\n";

$outfile="Update2005.txt";

print "outfile = $outfile\n";

$sql="

use $db;

update

cymgaspf02v3 c,

fuels2005.countyyearmonth2005 d,

fuels2005.gasoline2005 g

set

c.RVP =g.RVP,

c.GasSulfur =g.GasSulfur,

c.GasMaxSulfur =g.GasMaxSulfur,

c.EtohVolume =g.EtohVolume,

c.EtohMktShare =g.EtohMktShare,

c.MtbeVolume =g.MtbeVolume,

c.MtbeMktShare =g.MtbeMktShare,

c.EtbeVolume =g.EtbeVolume,

c.EtbeMktShare =g.EtbeMktShare,

c.TameVolume =g.TameVolume,

c.TameMktShare =g.TameMktShare,

c.AromaticContent =g.AromaticContent,

c.OlefinContent =g.OlefinContent,

c.BenzeneContent =g.BenzeneContent,

c.E200 =g.E200,

c.E300 =g.E300

where

d.year=2005

and c.year=2005

and c.fipsstateid=d.fipsstateid

and c.fipscountyid=d.fipscountyid

and c.month=d.month

and d.hwygasolineid=g.gasolineid

;

# Test

# Old one to compare to

select

'cymgasncd20070123' as tbl,

year,

rfg,

count(rvp),

avg(rvp),

sum(rvp)

from cymgasncd20070123

group by year,rfg

order by year,rfg

;

# New one to compare to

select

'cymgaspf02v3' as tbl,

year,

rfg,

count(rvp),

avg(rvp),

sum(rvp)

from cymgaspf02v3

group by year,rfg

order by year,rfg

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

#`mysql < script.sql`;

`mysql < script.sql > $outfile`;

### Script: Update2009.plx

# Update2009.plx

# Substitute Dave Brzezinski's 2009 table for 2009.

# Source: Fuels2009.CountyYearMonthRFS2009Fix and Gasolinerfs2009Fix

$db="fuelspf02v3";

print "db = $db\n";

$outfile="Update2009.txt";

print "outfile = $outfile\n";

$sql="

use $db;

update

cymgaspf02v3 c,

fuels2009.countyyearmonthrfs2009fix d,

fuels2009.gasolinerfs2009fix g

set

c.RVP =g.RVP,

c.GasSulfur =g.GasSulfur,

c.GasMaxSulfur =g.GasMaxSulfur,

c.EtohVolume =g.EtohVolume,

c.EtohMktShare =g.EtohMktShare,

c.MtbeVolume =g.MtbeVolume,

c.MtbeMktShare =g.MtbeMktShare,

c.EtbeVolume =g.EtbeVolume,

c.EtbeMktShare =g.EtbeMktShare,

c.TameVolume =g.TameVolume,

c.TameMktShare =g.TameMktShare,

c.AromaticContent =g.AromaticContent,

c.OlefinContent =g.OlefinContent,

c.BenzeneContent =g.BenzeneContent,

c.E200 =g.E200,

c.E300 =g.E300

where

d.year=2009

and c.year=2009

and c.fipsstateid=d.fipsstateid

and c.fipscountyid=d.fipscountyid

and c.month=d.month

and d.hwygasolineid=g.gasolineid

;

# Test

# Old one to compare to

select

'cymgasncd20070123' as tbl,

year,

rfg,

count(rvp),

avg(rvp),

sum(rvp)

from cymgasncd20070123

group by year,rfg

order by year,rfg

;

# New one to compare to

select

'cymgaspf02v3' as tbl,

year,

rfg,

count(rvp),

avg(rvp),

sum(rvp)

from cymgaspf02v3

group by year,rfg

order by year,rfg

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

#`mysql < script.sql`;

`mysql < script.sql > $outfile`;

### Script: InterpolateFuelsAllMonths.plx. Joins cymgaspf02v3 to itself twice and interpolates.

# InterpolateFuelsAllMonths.plx

# Interpolate all months between two years.

# Note that:

# PR and VI are being treated the same as all states.

# mtbe, etbe, and tame are being interpolated the same as etoh

# For 2004, interpolate between 2003 and 2005

# For 2006, interpolate between 2005 and 2009

foreach $yearyear1year2 (

#'2004 2003 2005',

#'2006 2005 2009',

#'2007 2005 2009',

#'2008 2005 2009',

'2010 2009 2012',

'2011 2009 2012',

) {

($year,$year1,$year2)=split(/ /,$yearyear1year2);

# $year is the year to produce by interpolation between $year1 & $year2.

print "year = $year\n";

print "year1 = $year1\n";

print "year2 = $year2\n";

$mult=($year-$year1)/($year2-$year1);

print "mult = $mult\n";

$db="fuelsPF02v3";

print "db = $db\n";

$table="cymgaspf02v3";

print "table = $table\n";

$outfile="InterpolateFuelsAllMonths$year.txt";

print "outfile = $outfile\n";

$sql="

use $db

;

# Interpolate fuel properties between all months

update

$table a,$table b,$table c

set

a.rvp=b.rvp+(c.rvp-b.rvp)\*$mult,

a.gassulfur=b.gassulfur+(c.gassulfur-b.gassulfur)\*$mult,

a.gasmaxsulfur=b.gasmaxsulfur+(c.gasmaxsulfur-b.gasmaxsulfur)\*$mult,

a.aromaticcontent=b.aromaticcontent+(c.aromaticcontent-b.aromaticcontent)\*$mult,

a.olefincontent=b.olefincontent+(c.olefincontent-b.olefincontent)\*$mult,

a.benzenecontent=b.benzenecontent+(c.benzenecontent-b.benzenecontent)\*$mult,

a.e200=b.e200+(c.e200-b.e200)\*$mult,

a.e300=b.e300+(c.e300-b.e300)\*$mult,

a.etohmktshare=b.etohmktshare+(c.etohmktshare-b.etohmktshare)\*$mult,

a.etohvolume=((b.etohmktshare\*b.etohvolume)+(c.etohmktshare\*c.etohvolume-b.etohmktshare\*b.etohvolume)\*$mult)/a.etohmktshare,

a.mtbemktshare=b.mtbemktshare+(c.mtbemktshare-b.mtbemktshare)\*$mult,

a.mtbevolume=((b.mtbemktshare\*b.mtbevolume)+(c.mtbemktshare\*c.mtbevolume-b.mtbemktshare\*b.mtbevolume)\*$mult)/a.mtbemktshare,

a.etbemktshare=b.etbemktshare+(c.etbemktshare-b.etbemktshare)\*$mult,

a.etbevolume=((b.etbemktshare\*b.etbevolume)+(c.etbemktshare\*c.etbevolume-b.etbemktshare\*b.etbevolume)\*$mult)/a.etbemktshare,

a.tamemktshare=b.tamemktshare+(c.tamemktshare-b.tamemktshare)\*$mult,

a.tamevolume=((b.tamemktshare\*b.tamevolume)+(c.tamemktshare\*c.tamevolume-b.tamemktshare\*b.tamevolume)\*$mult)/a.tamemktshare

where

a.year=$year

and b.year=$year1

and c.year=$year2

and a.month=b.month

and a.month=c.month

and a.fipsstateid=b.fipsstateid

and a.fipsstateid=c.fipsstateid

and a.fipscountyid=b.fipscountyid

and a.fipscountyid=c.fipscountyid

;

# Test

# Old one to compare to

select

'cymgasncd20070123' as tbl,

year,

rfg,

count(rvp),

avg(rvp),

sum(rvp)

from cymgasncd20070123

group by year,rfg

order by year,rfg

;

# New one to compare to

select

'cymgaspf02v3' as tbl,

year,

rfg,

count(rvp),

avg(rvp),

sum(rvp)

from cymgaspf02v3

group by year,rfg

order by year,rfg

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql > $outfile`;

}

### Script: Set20072008MTBEZero.plx

# Set20072008MTBEZero.plx

$db="fuelspf02v3";

print "db = $db\n";

$outfile="Set20072008MTBEZero.txt";

print "outfile = $outfile\n";

$sql="

use $db;

update cymgaspf02v3

set

MtbeVolume=0.0,

MtbeMktShare=0.0

where year in(2007,2008)

;

# Test

# Old one to compare to

select

'cymgasncd20070123' as tbl,

year,

avg(mtbevolume),

avg(mtbemktshare),

sum(mtbevolume),

sum(mtbemktshare)

from cymgasncd20070123

group by year

order by year

;

# New one to compare to

select

'cymgaspf02v3' as tbl,

year,

avg(mtbevolume),

avg(mtbemktshare),

sum(mtbevolume),

sum(mtbemktshare)

from cymgaspf02v3

group by year

order by year

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

#`mysql < script.sql`;

`mysql < script.sql > $outfile`;

### Script: Update20122050.plx

# Update20122050.plx

# Substitute Fuels2020 for all years from 2012 to 2050

# Source: Fuels2020.CountyYearMonthRfsMsat2020 and GasolinerfsMsat2020

$db="fuelspf02v3";

print "db = $db\n";

$outfile="Update20122050.txt";

print "outfile = $outfile\n";

$sql="

use $db;

update

cymgaspf02v3 c,

fuels2020.countyyearmonthrfsmsat2020 d,

fuels2020.gasolinerfsmsat2020 g

set

c.RVP =g.RVP,

c.GasSulfur =g.GasSulfur,

c.GasMaxSulfur =g.GasMaxSulfur,

c.EtohVolume =g.EtohVolume,

c.EtohMktShare =g.EtohMktShare,

c.MtbeVolume =g.MtbeVolume,

c.MtbeMktShare =g.MtbeMktShare,

c.EtbeVolume =g.EtbeVolume,

c.EtbeMktShare =g.EtbeMktShare,

c.TameVolume =g.TameVolume,

c.TameMktShare =g.TameMktShare,

c.AromaticContent =g.AromaticContent,

c.OlefinContent =g.OlefinContent,

c.BenzeneContent =g.BenzeneContent,

c.E200 =g.E200,

c.E300 =g.E300

where

d.year=2020

and c.year>=2012

and c.fipsstateid=d.fipsstateid

and c.fipscountyid=d.fipscountyid

and c.month=d.month

and d.hwygasolineid=g.gasolineid

;

# Test

# Old one to compare to

select

'cymgasncd20070123' as tbl,

year,

rfg,

avg(rvp),

avg(benzenecontent)

from cymgasncd20070123

group by year,rfg

order by year,rfg

;

# New one to compare to

select

'cymgaspf02v3' as tbl,

year,

rfg,

avg(rvp),

avg(benzenecontent)

from cymgaspf02v3

group by year,rfg

order by year,rfg

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

#`mysql < script.sql`;

`mysql < script.sql > $outfile`;

### Script: Set2011BenzTo2012.plx

# Set2011BenzTo2012.plx

$db="fuelsPF02v3";

print "db = $db\n";

$table="cymgaspf02v3";

print "table = $table\n";

$outfile="Set2011BenzTo2012.txt";

print "outfile = $outfile\n";

$sql="

use $db

;

# Interpolate fuel properties between all months

update

$table a,$table b

set

a.benzenecontent=b.benzenecontent

where

a.year=2011

and b.year=2012

and a.month=b.month

and a.fipsstateid=b.fipsstateid

and a.fipscountyid=b.fipscountyid

;

# Test

# Old one to compare to

select

'cymgasncd20070123' as tbl,

year,

avg(rvp),

avg(benzenecontent)

from cymgasncd20070123

group by year

order by year

;

# New one to compare to

select

'cymgaspf02v3' as tbl,

year,

avg(rvp),

avg(benzenecontent)

from cymgaspf02v3

group by year

order by year

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql > $outfile`;

### Script: FuelsToNmimAllYears.plx

# FuelsToNmimAllYears.plx

# This script is for the Pf02v3 fuels.

# the cymgas table is cymgaspf02v3.

# Starts with a table converted from CYMGasPf02v3

# that contains both countyyearmonth data and gasoline data.

# Splits table with counties and fuels properties

# into CountyYearMonthPf02v3 and GasolinePf02v3 tables for a single year.

# Script to populate NMIM files $cym and $gas from $cymgas

# This is the last step in creating the NMIM fuels tables from

# $cymgas

$db="FuelsPf02v3";

print "db = $db\n";

$tblvers="Pf02v3";

print "tblvers = $tblvers\n";

$cymgas="$db.cymgasPf02v3";

print "cymgas = $cymgas\n";

$cym="CountyYearMonth$tblvers";

print "cym = $cym\n";

$gas="Gasoline$tblvers";

print "gas = $gas\n";

$outfile="${cym}Test.txt";

print "outfile = $outfile\n";

$sql="

USE $db;

#drop table if exists $gas;

#CREATE TABLE $gas (

# RVP float NOT NULL default '0',

# GasSulfur float NOT NULL default '0',

# GasMaxSulfur float NOT NULL default '0',

# ETOHVolume float NOT NULL default '0',

# ETOHMktShare float NOT NULL default '0',

# MTBEVolume float NOT NULL default '0',

# MTBEMktShare float NOT NULL default '0',

# ETBEVolume float NOT NULL default '0',

# ETBEMktShare float NOT NULL default '0',

# TAMEVolume float NOT NULL default '0',

# TAMEMktShare float NOT NULL default '0',

# AromaticContent float NOT NULL default '0',

# OlefinContent float NOT NULL default '0',

# BenzeneContent float NOT NULL default '0',

# E200 float NOT NULL default '0',

# E300 float NOT NULL default '0',

# RFG char(1) NOT NULL default ''

# )TYPE=MyISAM ;

#

#

###Populate $gas from $cym

#

#INSERT $gas

# SELECT

# RVP,

# GasSulfur,

# GasMaxSulfur,

# ETOHVolume,

# ETOHMktShare,

# MTBEVolume,

# MTBEMktShare,

# ETBEVolume,

# ETBEMktShare,

# TAMEVolume,

# TAMEMktShare,

# AromaticContent,

# OlefinContent,

# BenzeneContent,

# E200,

# E300,

# RFG

# FROM $cymgas

# group by

# RVP,

# GasSulfur,

# GasMaxSulfur,

# ETOHVolume,

# ETOHMktShare,

# MTBEVolume,

# MTBEMktShare,

# ETBEVolume,

# ETBEMktShare,

# TAMEVolume,

# TAMEMktShare,

# AromaticContent,

# OlefinContent,

# BenzeneContent,

# E200,

# E300,

# RFG

# order by

# RVP,

# GasSulfur,

# GasMaxSulfur,

# ETOHVolume,

# ETOHMktShare,

# MTBEVolume,

# MTBEMktShare,

# ETBEVolume,

# ETBEMktShare,

# TAMEVolume,

# TAMEMktShare,

# AromaticContent,

# OlefinContent,

# BenzeneContent,

# E200,

# E300,

# RFG

#;

#

###Add unique ID field to $gas and rvpoxywaiver

#

#ALTER TABLE $gas

# ADD (

# Gasolineid smallint(6) NOT NULL default 0 PRIMARY KEY AUTO\_INCREMENT,

# rvpoxywaiver tinyint not null default '1'

# )

#;

#ALTER TABLE $gas

#add index(RVP),

#add index(GasSulfur),

#add index(GasMaxSulfur),

#add index(ETOHVolume),

#add index(ETOHMktShare),

#add index(MTBEVolume),

#add index(MTBEMktShare),

#add index(ETBEVolume),

#add index(ETBEMktShare),

#add index(TAMEVolume),

#add index(TAMEMktShare),

#add index(AromaticContent),

#add index(OlefinContent),

#add index(BenzeneContent),

#add index(E200),

#add index(E300)

#;

#

## Diagnostic

## select \* from $gas;

#

#

## Add gasolineid field to $cymgas

##ALTER TABLE $cymgas

## ADD Gasolineid smallint(6) NOT NULL default 0

##;

#

#

#UPDATE $cymgas as a, $gas as b

# SET a.gasolineid = b.gasolineid

# WHERE

# a.RVP = b.RVP and

# a.GasSulfur = b.GasSulfur and

# a.GasMaxSulfur = b.GasMaxSulfur and

# a.ETOHVolume = b.ETOHVolume and

# a.ETOHMktShare = b.ETOHMktShare and

# a.MTBEVolume = b.MTBEVolume and

# a.MTBEMktShare = b.MTBEMktShare and

# a.ETBEVolume = b.ETBEVolume and

# a.ETBEMktShare = b.ETBEMktShare and

# a.TAMEVolume = b.TAMEVolume and

# a.TAMEMktShare = b.TAMEMktShare and

# a.AromaticContent = b.AromaticContent and

# a.OlefinContent = b.OlefinContent and

# a.BenzeneContent = b.BenzeneContent and

# a.E200= b.e200 and

# a.e300 = b.e300 and

# a.RFG = b.RFG

#;

# $cym = CountyYearMonthPf02v3 already exists

# and contains correct variables.

## Replace GasolineIDs in $cym

#UPDATE $cym as a, $cymgas as b

# SET a.hwygasolineid=gasolineid, a.nrgasolineid=gasolineid

# WHERE

# a.fipsstateid = b.fipsstateid

# and a.fipscountyid = b.fipscountyid

# and a.year =b.year

# and a.month =b.month

#;

## Testing

## Verify the number of gasolineid's

select count(distinct hwygasolineid) from $cym;

select count(distinct gasolineid) from $gas;

# Test by getting some averages by county

select 'Average by county';

select avg(etohvolume),avg(etohmktshare),

avg(mtbevolume),avg(mtbemktshare),

avg(etbevolume),avg(etbemktshare),

avg(tamevolume),avg(tamemktshare),

avg(benzenecontent)

FROM

$cym as a,

$gas as b

where a.hwygasolineid=b.gasolineid

;

# Test by getting some averages by fuel

select 'Average by fuel';

select avg(etohvolume),avg(etohmktshare),

avg(mtbevolume),avg(mtbemktshare),

avg(etbevolume),avg(etbemktshare),

avg(tamevolume),avg(tamemktshare),

avg(benzenecontent)

FROM $gas

;

## Testing## retrieve results for 3 specific counties

## From $cym joined to $gas

select

'cymjoinedtogas' as src,

a.fipsstateid,

a.fipscountyid,

a.year,

a.month,

a.hwygasolineid,

b.gasolineid,

RVP,

GasSulfur,

GasMaxSulfur,

ETOHVolume,

ETOHMktShare,

MTBEVolume,

MTBEMktShare,

ETBEVolume,

ETBEMktShare,

TAMEVolume,

TAMEMktShare,

AromaticContent,

OlefinContent,

BenzeneContent,

E200,

E300,

RFG

FROM $cym as a, $gas as b

where a.hwygasolineid=b.gasolineid and

((a.fipsstateid = 17 and a.fipscountyid = 31) or

(a.fipsstateid = 10 and a.fipscountyid = 5) or

(a.fipsstateid = 16 and a.fipscountyid = 15))

and

(a.year = 2002 or a.year = 2011)

order by

fipsstateid,

fipscountyid,

year,

month

;

## From $cymgas

select

'$cymgas' as src,

fipsstateid,

fipscountyid,

year,

month,

0 as hwygasolineid,

gasolineid,

RVP,

GasSulfur,

GasMaxSulfur,

ETOHVolume,

ETOHMktShare,

MTBEVolume,

MTBEMktShare,

ETBEVolume,

ETBEMktShare,

TAMEVolume,

TAMEMktShare,

AromaticContent,

OlefinContent,

BenzeneContent,

E200,

E300,

RFG

FROM $cymgas

where

((fipsstateid = 17 and fipscountyid = 31) or

(fipsstateid = 10 and fipscountyid = 5) or

(fipsstateid = 16 and fipscountyid = 15))

and

(year = 2002 or year = 2011)

order by

fipsstateid,

fipscountyid,

year,

month

;

## Output the whole works -- too large

#select a.\*,b.\*

#FROM $cym as a, $gas as b

#where a.hwygasolineid=b.gasolineid

#;

FLUSH TABLES ;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql > $outfile`;

## Program end.

# ChangeLog for NCD20070123pf02v3

One change is to correct CountyMonthHour2005 table. It was missing Broomfield County, Colorado (08014).

The new table CountyMonthHour2005a was created.

Temp and Humidity data were added for Broomfield County by copying data from Boulder County, Colorado (08013).

use NCD20070123pf02v3;

create table countymonthhour2005a

select \* from countymonthhour2005;

insert into countymonthhour2005a

select '14',fipsstateid,month,hourID,AverageTemp,AverageRelHumidity from countymonthhour2005

where fipsstateid=8 and fipscountyid=13;

SELECT count(\*) FROM NCD20070123pf02v3.countymonthhour2005a c;

928224

SELECT count(\*) FROM NCD20070123pf02v3.countymonthhour2005 c;

927936

928224-927936=288

SELECT sum(c.AverageTemp), sum(c.AverageRelHumidity) FROM countymonthhour2005a c

WHERE ! (c.FIPSStateId=8 AND c.FIPSCountyId=14);

51058503.198778, 61735725.900726

SELECT sum(c.AverageTemp), sum(c.AverageRelHumidity) FROM countymonthhour2005 c;

51058503.198778, 61735725.900726

SELECT sum(c.AverageTemp), sum(c.AverageRelHumidity) FROM countymonthhour2005 c

WHERE c.FIPSStateId=8 AND c.FIPSCountyId=13;

13632.499998093, 14470.199998856

SELECT sum(c.AverageTemp), sum(c.AverageRelHumidity) FROM countymonthhour2005a c

WHERE c.FIPSStateId=8 AND c.FIPSCountyId=14;

13632.499998093, 14470.199998856

RFS Fuels were also included in special files for 2002, 2009, 2014, 2020, and 2030 with help from Rick Rykowski. Met data for 2005 was added to enable NEI2005 run. Baseyearvmtnew3 was used throughout, except for 2005, which was added from HPMS with input from OAQPS.

# ChangeLog for NCD20070123

The change is to correct chromium emission factors for onroad vehicles. Total chromium emissions remain the same, but the ratio of Cr6:Total has changed from 40% to 18%, as shown in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Old** | Diesel | Gas | gas/diesel |
| Cr3 | 7.9100E-07 | 2.9700E-06 | 3.7547E+00 |
| Cr6 | 5.2800E-07 | 1.9800E-06 | 3.7500E+00 |
| Total | 1.3190E-06 | 4.9500E-06 | 3.7528E+00 |
| % Cr6 | 40.03% | 40.00% |  |
|  |  |  |  |
| **New** | Diesel | Gas |  |
| Cr3 | 1.081580E-06 | 4.059000E-06 | 3.7528E+00 |
| Cr6 | 2.374200E-07 | 8.910000E-07 | 3.7528E+00 |
| Total | 1.319000E-06 | 4.950000E-06 | 3.7528E+00 |
| % Cr6 | 18.00% | 18.00% |  |

The script used for the conversion is as follows:

# Script to change onroad Chromium

# EFs adjusted so that Cr6+ is 18% and Cr3+ is 82% of total Cr.

# Pollutantcode id for Cr6+ is 12

# Pollutantcode id for Cr3+ is 47

$ncd='ncd20070123';

print "ncd = $ncd\n";

$sql="

use $ncd;

update scctoxics t, scc s set

exhDiesel=1.08158E-06

where

pollutantcodeid=47

and t.sccid=s.sccid

and t.sccid<=144

and fueltype='Dies'

;

update scctoxics t, scc s set

exhBaseGas=4.059E-06,

exhEthGas=4.059E-06,

exhMTBEGas=4.059E-06,

exhRFGGas=4.059E-06

where

pollutantcodeid=47

and t.sccid=s.sccid

and t.sccid<=144

and fueltype='Gas'

;

update scctoxics t, scc s set

exhDiesel=2.3742E-07

where

pollutantcodeid=12

and t.sccid=s.sccid

and t.sccid<=144

and fueltype='Dies'

;

update scctoxics t, scc s set

exhBaseGas=8.91E-07,

exhEthGas=8.91E-07,

exhMTBEGas=8.91E-07,

exhRFGGas=8.91E-07

where

pollutantcodeid=12

and t.sccid=s.sccid

and t.sccid<=144

and fueltype='Gas'

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

# ChangeLog for NCD20061227

In RFSFinal runs, EPAME.nlv external file M6 complained about lines not adding up to 1.

Correction to EPAME.nlv is as follows:

LDGV, T1, line 6 from 0.03 to 0.30.

LDGT2, T1, line 6, from 0.03 to 0.30

LDGT2, TLEV, line 7, from 0.4 to 0.38

# ChangeLog for NCD20061219

Pollutantcode and scctoxics tables revised to add mercury and arsenic. Scripts are below, and include emissions factors. Emission factors were provided by Marion Hoyer in email

## Marion Hoyer email 11/6/06

Marion Hoyer/AA/USEPA/US

EPA-OAR,OTAQ,ASD

Sent by: Marion Hoyer

Received Date:

11/6/2006 13:23

Transmission Date:

11/6/2006 13:23

To

Harvey Michaels/AA/USEPA/US@EPA

cc

Carl Scarbro/AA/USEPA/US@EPA, Michael Christianson/AA/USEPA/US@EPA,

Richard Baldauf/RTP/USEPA/US@EPA, Rich Cook/AA/USEPA/US@EPA, Kathryn

Sargeant/AA/USEPA/US@EPA, Laurel Driver/RTP/USEPA/US@EPA

Subject

some changes to Hg and As EFs

Hello Harvey,

This file has the final emission factors for arsenic and the three forms

of mercury.

I made some changes after last Thursday so you can delete that file.

Carl and I are working on the document that describes how these emission

factors were developed. We'll send that around when it is ready for

review.

Let me know if you have any questions.

Marion

[attachment "Final Emission Factors for Mercury and Arsenic 2006.xls"

deleted by Harvey Michaels/AA/USEPA/US]

## Script: AddMetalSCCToxics.plx

# AddMetalSCCToxics.plx

# Script to add Metals to SCCToxics table

# Metals do not have entries for LPG and CNG SCCs

# $pid = pollutantcodeid

# $ogef = onroad gasoline emission factor

# $odef = onroad diesel emission factor

# $orat = onroad ratiotype

# $ngef = nonroad gasoline emission factor

# $ndef = nonroad diesel emission factor

# $nrat = nonroad ratiotype

$ncd='ncd20061201';

print "ncd = $ncd\n";

foreach $plt(

'67 1.06500E-07 6.2000E-09 Mile 1.77130E-06 1.200E-07 Gal',

'68 9.90000E-09 3.2000E-09 Mile 1.68300E-07 6.220E-08 Gal',

'69 4.00000E-10 1.6000E-09 Mile 6.90000E-09 3.210E-08 Gal',

'70 2.28000E-06 2.2800E-06 Mile 4.097826E-05 3.886598E-05 PM10',

) {

print "plt= $plt\n";

($pid,$ogef,$odef,$orat,$ngef,$ndef,$nrat)=split(/ /,$plt); # separator is tab

print "pid = $pid\n";

print "ogef = $ogef\n";

print "odef = $odef\n";

print "orat = $orat\n";

print "ngef = $ngef\n";

print "ndef = $ndef\n";

print "nrat = $nrat\n";

$sql="

use $ncd;

# use manganese as template

drop table if exists pid$pid;

create table pid$pid

select \* from scctoxics where pollutantcodeid=43

;

# onroad gas

update pid$pid p,scc s

set

p.pollutantcodeid='$pid',

exhBaseGas=$ogef,

exhEthGas=$ogef,

exhMTBEGas=$ogef,

exhRFGGas=$ogef,

exhDiesel=NULL,

evapBaseGas=NULL,

evapEthGas=NULL,

evapMTBEGas=NULL,

evapRFGGas=NULL,

ratioType='$orat'

where

p.sccid=s.sccid

and ornr='OR'

and fueltype='Gas'

;

# onroad diesel

update pid$pid p,scc s

set

p.pollutantcodeid='$pid',

exhBaseGas=NULL,

exhEthGas=NULL,

exhMTBEGas=NULL,

exhRFGGas=NULL,

exhDiesel=$odef,

evapBaseGas=NULL,

evapEthGas=NULL,

evapMTBEGas=NULL,

evapRFGGas=NULL,

ratioType='$orat'

where

p.sccid=s.sccid

and ornr='OR'

and fueltype='Dies'

;

# nonroad gas

update pid$pid p,scc s

set

p.pollutantcodeid='$pid',

exhBaseGas=$ngef,

exhEthGas=$ngef,

exhMTBEGas=$ngef,

exhRFGGas=$ngef,

exhDiesel=NULL,

evapBaseGas=NULL,

evapEthGas=NULL,

evapMTBEGas=NULL,

evapRFGGas=NULL,

ratioType='$nrat'

where

p.sccid=s.sccid

and ornr='NR'

and fueltype='Gas'

;

# nonroad diesel

update pid$pid p,scc s

set

p.pollutantcodeid=$pid,

exhBaseGas=NULL,

exhEthGas=NULL,

exhMTBEGas=NULL,

exhRFGGas=NULL,

exhDiesel=$ndef,

evapBaseGas=NULL,

evapEthGas=NULL,

evapMTBEGas=NULL,

evapRFGGas=NULL,

ratioType='$nrat'

where

p.sccid=s.sccid

and ornr='NR'

and fueltype='Dies'

;

# Add to scctoxicstable

insert into scctoxics (

sccid,

pollutantcodeid,

exhBaseGas,

exhEthGas,

exhMTBEGas,

exhRFGGas,

exhDiesel,

evapBaseGas,

evapEthGas,

evapMTBEGas,

evapRFGGas,

ratioType

)

select

sccid,

pollutantcodeid,

exhBaseGas,

exhEthGas,

exhMTBEGas,

exhRFGGas,

exhDiesel,

evapBaseGas,

evapEthGas,

evapMTBEGas,

evapRFGGas,

ratioType

from pid$pid

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

}

## Script: AddPollutantCode.plx

# AddPollutantCode.plx

# Script to add pollutants to pollutantcode

# $pid = pollutantcodeid

# $pcode = pollutantcode

# $pname = pollutantname

# $rspi = runspecpollutantindex

$ncd='ncd20061201';

print "ncd = $ncd\n";

foreach $pidpcodepnamerspi (

'67 200 Mercury (elemental gaseous) 21',

'68 201 Mercury (divalent gaseous) 21',

'69 202 Mercury (particulate) 21',

'70 93 Arsenic & compounds 21',

) {

print "pidpcodepnamerspi = $pidpcodepnamerspi\n";

($pid,$pcode,$pname,$rspi)=split(/ /,$pidpcodepnamerspi); # separator is tab

print "pid = $pid\n";

print "pcode = $pcode\n";

print "pname = $pname\n";

print "rspi = $rspi\n";

$sql="

use $ncd;

insert pollutantcode

(pollutantcodeid,pollutantcode,pollutantname,runspecpollutantindex)

values($pid,'$pcode','$pname',$rspi)

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

}

# ChangeLog for NCD20061218

Author: Dave Brzezinski

We reviewed the inputs used for implementation of California standards, which begin in the 2004 model year (LEV2) used in NMIM. A recent survey by EPA Region 9 identified 11 states with LEV2 programs:

California(6), Connecticut(9), Maine(23), Massachusetts(25), New Jersey(34), New York(36), Oregon(41), Pennsylvania(42), Rhode Island(44), Vermont(50) and Washington(53).

All of these programs begin with model year 2004, except Connecticut, Pennsylvania and Rhode Island, which revert to federal (Tier2) standards in the 2004 model year begin with California standards with the 2008 model year.

Fourteen states have LEV programs, either as part of the Northeast ozone transport regions (OTC) or as a result of state legislation. These states are:

California(6), Connecticut(9), Delaware(10), District of Columbia(11), Maine(23), Maryland(24), Massachusetts(25), New Hampshire(33), New Jersey(34), New York(36), Pennsylvania(42), Rhode Island(44), Vermont(50) and Virginia(51).

Since Oregon(41) and Washington(53) did not have LEV programs, a LEV phase in file (nlv) was added to the database to allow for the implementation of ZEVs for the LEV2 program. The LEV phase in portion of the phase in file is the default federal NLEV phase in and does not affect those model years. A single input file was created to be used by both states. The file name is:

EPAORWA.nlv

The California ZEV fractions used in the LEV phase in files are taken from the EPA guidance document, "Modeling Alternative NLEV Implementation and Adoption of California Standards in MOBILE6," (February 2, 2005) that was distributed to all EPA Regional offices.

The LEV phase in files (nlv) for Connecticut(9), Pennsylvania(42) and Rhode Island(44) are identical, so a single input file was created to be used by all three states. This file assumes no California ZEVs until the 2008 model year, when these states switch from federal Tier2 to California standards.

EPACTPAR.nlv

The LEV phase in files (nlv) for California(6), Maine(23), Massachusetts(25), New Jersey(34) and Vermont(50) all begin using California Standards in the 2004 model year and all use the same phase in values for the 2004 and later model years which include California ZEVs. The LEV phase in for each of these states is different, so each state will have it's own LEV phase in file. Since these files have been altered by EPA, we have removed the state submission file name convention, and given these files an EPA name:

EPACA.nlv

EPAME.nlv

EPAMA.nlv

EPANJ.nlv

EPAVT.nlv

A change was made in the LEV phase in file (nlv) for Massachusetts, which changes the mix for LEV vehicles for the 2003 model year which reduces the assumed number of ZEVs to EPA's most recent estimate. (0.703/0.293/0.004 for LEV/ULEV/ZEV)

The LEV phase in file (nlv) for Vermont already assumes zero ZEVs for the 2003 model year. However, the mix of LEVs and ULEVs was changed to match the more recent EPA estimates (0.705/0.295), with zero ZEVs.

The LEV phase in file (nlv) for New York(36) was changed to the mix of LEVs and ULEVs to match the more recent EPA estimates (0.705/0.295), with zero ZEVs. Since New York does not claim ZEV benefits, the fraction of ZEVs in all model years was left at zero.

EPANY.nlv

The LEV phase in files (nlv) for Delaware(10), District of Columbia(11), Maryland(24), New Hampshire(33) and Virginia(51) do not need to be changed, since these states are reverting to the default federal program (Tier2) beginning with model year 2004 instead of continuing to use California Standards.

10000.nlv (Delaware)

11001.nlv (District of Columbia)

24000.nlv (Maryland)

33000.nlv (New Hampshire)

51000.nlv (Virginia)

There are currently only two phase in cases for California Standards beginning with the 2004 model year (LEV2). Some states begin using California Standards with the 2004 model year (as in California) and some states begin with the 2008 model year. Therefore, we have created two sets of input files for Tier2 phase in that reflect the two cases. The states that begin with the 2004 model year are California(6), Maine(23), Massachusetts(25), New Jersey(34), New York(36), Oregon(41), Vermont(50) and Washington(53). The Tier2 phase-in files used by these states will be the same files as used for California:

EPACA.t2v

EPACA.t2c

EPACA.t2x

Currently, three states will begin California Standards with the 2008 model year. These states are Connecticut(9), Pennsylvania(42) and Rhode Island(44). The Tier2 phase-in files used by these states are:

EPA2008.t2v

EPA2008.t2c

EPA2008.t2x

A copy of these new external files have been added to the externalfiles directory of the MySQL NCD. The following external files have been superceded and have been deleted from the externalfiles directory:

6000.nlv

09000.nlv

23000.nlv

25000.nlv

25000.t2x

25000.t2v

25000.t2c

34000.nlv

36000.nlv

36000.t2x

36000.t2v

36000.t2c

42000.nlv

44000.nlv

50000.nlv

# Batch file to Delete external files that are not relevant any more

cd C:\mysql\data\NCD20061218\ExternalFiles

DEL 6000.nlv 09000.nlv 23000.nlv 25000.nlv 25000.t2x 25000.t2v 25000.t2c

DEL 34000.nlv 36000.nlv 36000.t2x 36000.t2v 36000.t2c 42000.nlv 44000.nlv 50000.nlv

# MySQL script to update the STATE table.

# Oregon and Washington:

update state set nlevfilename="EPAORWA.nlv"

where fipsstateid in (41,53);

# Connecticut, Pennsylvania and Rhode Island:

update state set nlevfilename="EPACTPAR.nlv"

where fipsstateid in (9,42,44);

# California:

update state set nlevfilename="EPACA.nlv"

where fipsstateid in (6);

# Maine:

update state set nlevfilename="EPAME.nlv"

where fipsstateid in (23);

# Massachusetts:

update state set nlevfilename="EPAMA.nlv"

where fipsstateid in (25);

# New Jersey:

update state set nlevfilename="EPANJ.nlv"

where fipsstateid in (34);

# New York:

update state set nlevfilename="EPANY.nlv"

where fipsstateid in (36);

# Vermont:

update state set nlevfilename="EPAVT.nlv"

where fipsstateid in (50);

# LEV2 programs that begin with the 2004 model year:

# CA, ME, MA, NJ, NY, OR, VT and WA.

update state

set t2exhphaseinfilename="EPACA",

t2evapphaseinfilename="EPACA",

t2certfilename="EPACA"

where fipsstateid in (6,23,25,34,36,41,50,53);

# LEV2 programs that begin with the 2008 model year:

# CT, PA and RI.

update state

set t2exhphaseinfilename="EPA2008",

t2evapphaseinfilename="EPA2008",

t2certfilename="EPA2008"

where fipsstateid in (9,42,44);

# ChangeLog for NCD20061128

## Background and general notes:

The changes are designed to make nonroad files submitted by states as part of the 2002 NEI consistent with NONROAD2005. This was necessary because many state submitted files which altered only a few lines, leaving the others consistent with NONROAD2004.

## Craig Harvey's Final Email Tranmitting finished files

Craig Harvey/AA/USEPA/US

EPA-OAR,OTAQ,ASD

Received Date:

11/27/2006 02:02 PM

Transmission Date:

11/27/2006 02:02:48 PM

To

Harvey Michaels/AA/USEPA/US@EPA

cc

Lyubov Mikhaylova/AA/USEPA/US@EPA

bcc

Subject

Re: NR External Files

Here are the fully updated, fixed, enhanced external POP files. Note that

besides the rec marine Hp distribution changes, I also fixed snowmobiles

in Rhode Island (44000.pop), since there were multiple problems with them

(including a tab character instead of a space). And I moved all the hydro

power unit lines to the same location they are in the NR05 pop files (per

their new SCC), which is more of a cosmetic and future ease of maintenance

issue, rather than a necessary change.

There is no 19000.pop file, because that should be removed from the

countyyear table as an external file (and the old one deleted from the

MySQL NCD ExternalFiles directory). And as mentioned initially, the same

deletions should be done to the "xx000.act" files (leaving only the

"xx00002.act" files).

[attachment "NewPop2.zip" deleted by Harvey Michaels/AA/USEPA/US]

Craig

## Luba Mikhaylova's Notes:

1. External files updated to be consistent with NR05.

* 19000.pop deleted as it is exactly the same as Default file.
* 18000.act, 26000.act, 39000.act, 39000.act,55000.act deleted since NMIM does not use them.
* <FIPSStateID>00002.act (activity) files updated according to NR05 changes.
* GRW(growth) files changed in Scrappage section to more detailed NR05 version. (SCRAPPAGE section copied from default NR05 files into state submitted files).
* POP (equip population) files fixed, updated and enhanced to be consistent with NR05.
* SEASON files: since the state data that was changed from NR04 should be good, and there is no "legacy" NR04 data that isn't the same as NR05. (except the US Total section, but that wouldn't be used by NMIM) – Craig Harvey

**2.**CountyNRFiles table:

* Records with19000.pop deleted from countynrfile table.

Script:

use ncd20061128;

delete from countynrfile

where CountyNRFileName='19000';

## Perl Scripts:

1. # Script to identify lines that are different and that are identical

# in state submitted \*.act files and DEFAULT NR04 ACTIVITY.DAT

foreach $stateid

('1700002', '1800002','1900002','2600002','2700002',,'3900002',,'5500002'){

$infile="oldExternalFiles\\${stateid}.act";

open (in1,$infile);

$infile2="nr04\\ACTIVITY.DAT";

open (in2,"$infile2");

$outfile1="${stateid}actDiff.txt";

open(out1,">>$outfile1");

$outfile2="${stateid}actIdentical.txt";

open (out2, ">>$outfile2");

$start=0; $i=0; $scc = 0;

while (<in2>) {

if($\_ =~ m/ACTIVITY/) {

@lines2 = <in2>;}

}

$i=0;

while (<in1>) {

if($\_ =~m/END/) {

$start=0; }

if( $start==1) {

$curLine = $\_;

$scc = substr($curLine,0,10);

$loadfactor=substr($curLine,76,5);

$activitylevel=substr($curLine,96,10);

$stLine= $lines2[$i];

$stScc = substr($stLine,0,10);

$stLF =substr($stLine,76,5);

$stAL=substr($stLine,96,10);

if($scc==$stScc) {

if($loadfactor==$stLF &&

$activitylevel == $stAL) {

print out2 $lines2[$i];

}

else {

print out1 $curLine;

}

}

$i++;

}

if($\_ =~ m/ACTIVITY/) {

$start=1;

$i=0;

}

}

print "Lines compared $i\n";

close (out1);

close (out2);

}

==========================================================================

#Srcipt to create new \*.act files using lines from state submitted files that are different from default NR04 #ACTIVITY.DAT and substituting NR04 identical lines with lines from NR05 ACTIVITY.DAT

%ID = ( "DE", 10000,

"IA", 19000,

"IL", 17000,

"IN", 18000,

"MI", 26000,

"OH", 39000,

"RI", 44000,

"WI", 55000 );

foreach $stateid ('1700002',,'1800002','1900002',

'2600002','2700002', '3900002','5500002'){

$infile1 = "nr05\\ACTIVITY.DAT";

open (in1,$infile1);

$infile2="${stateid}actDiff.txt";

$outfile = "newExternalFiles\\${stateid}.act";

open(out1,">>$outfile");

$start=0;

$linefound=0;

$curscc = -1;

$i=0;

while (<in1>) {

if($\_ =~ m/ACTIVITY/) {

$start=$k;

print out1 $\_;

}

if($\_ =~m/END/) {

$start=0;

}

if($start==0) {

print out1 $\_;

}

if($start >0 && $k>$start) { #

$i++;

$curline = $\_;

$curscc = substr($curline, 0,10);

$linefound=0;

open (in2,$infile2);

while (<in2>) {

unless ($linefound ==1) {

$difline = $\_;

$difscc = substr($difline, 0,10);

if($difscc==$curscc) {

$linefound=1;

}

}

} #while in2

if($linefound == 1) {

print out1 $difline;

}

else {

print out1 $curline;

}

} #if start

$k++;

}# while in1

close (out1);

} #foreach

===============================================================================

1. # Script to identify lines that are different and that are identical

# in state submitted \*.pop files and DEFAULT NR04 \*.POP

%ID = ( "DE", 10000,

"IA", 19000,

"IL", 17000,

"IN", 18000,

"MI", 26000,

"OH", 39000,

"RI", 44000,

"WI", 55000 );

foreach $state ('DE','IA','IL','IN','MI','OH','RI','WI')

{

$stateid = $ID{$state};

$infile="oldExternalFiles\\${stateid}.pop";

open (in1,$infile);

$infile2="newPOP\\${state}.pop";

open (in2,"$infile2");

$outfile1="${state}popDiff.txt";

open(out1,">>$outfile1");

$outfile2="${state}popIdentical.txt";

open (out2, ">>$outfile2");

$start=0;

$i=0;

$scc = 0;

while (<in2>) {

if($\_ =~ m/POPULATION/) {

@lines2 = <in2>;

}

}

$i=0;

while (<in1>) {

if($\_ =~m/END/) {

$start=0;

}

if( $start==1) {

$curLine = $\_;

@line=split(/\s/, $\_);

$scc = $line[8];

if($curLine eq $lines2[$i]) {

print out2 "$curLine "; }

else {

print out1 "$curLine";

}

$i++;

}

if($\_ =~ m/POPULATION/) {

$start=1;

$i=0;

}

}

print "Lines compared $i\n";

close (out1);

close (out2);

}

#Srcipt to create new \*.pop files using lines from state submitted files that are different from default NR04 #<StateAbbr>.POP and substituting NR04 identical lines with lines from NR05 #<StateAbbr>.POP

%ID = ( "DE", 10000,

"IA", 19000,

"IL", 17000,

"IN", 18000,

"MI", 26000,

"OH", 39000,

"RI", 44000,

"WI", 55000 );

foreach $state ('DE', 'IA' ,'IL','IN','MI','OH','RI','WI') #IA has no differences

{

$stateid = $ID{$state};

$infile1 = "oldExternalFiles\\${stateid}.pop";

open (in1,$infile1);

$infile2="${state}popDiff.txt";

$infile3="NR05POP\\${state}.pop";

$outfile = "newExternalFiles\\${stateid}.pop";

print "outfile $outfile\n";

open(out1,">>$outfile");

$start=0;

$scc=0;

$linefound=0;

$curscc = -1;

$i=0;

while (<in1>) {

if($\_ =~ m/POPULATION/) {

$start=$k;

print out1 $\_;

}

if($\_ =~m/END/) {

$start=0;

}

if($start==0) {

print out1 $\_;

}

if($start >0 && $k>$start) {

$i++;

$curline = $\_;

$curscc = substr($curline, 18,9);

if($curscc == $scc) {

if($linefound==1) {

print out1 $curline;

}

}

else { #

$scc = $curscc;

$linefound=0;

open (in2,$infile2);

while (<in2>) {

unless ($linefound ==1) {

$difline = $\_;

$difscc = substr($difline, 18,9);

if($difscc==$scc) {

$linefound=1;

}

}

} #while in2

if($linefound == 1) {

print out1 $curline;

if($i<12) {

print "linefound = $linefound\n";

}

}

else {

open (in3, $infile3);

$j=0;

$start1=0;

while (<in3>) {

$newline = $\_;

$n=0;

if($newline =~ m/POPULATION/) {

$start1=$j;

$n=1;

}

if($newline =~m/END/) {

$start1=0;

}

if($start1>0 && $j>$start1){

$n++;

$newscc = substr($newline, 18,9);

if ($newscc == $scc) {

print out1 $newline;

if($i<12) {

print "newline $newline";

}

}

}

$j++;

}#while in3

}#else

}#else

} #if start

$k++;

}# while in1

} #foreach

===============================================================================

# ChangeLog for NCD20060725

**1**. External Files: NLEV files updated to begin Tier 2 standards in 2004. djb 20051214

This is now a standard external data files. Files updated are:

10000.nlv, 11001.nlv, 23000.nlv, 24000.nlv, 25000.nlv, 36000.nlv, 42000.nlv,

44000.nlv, 50000.nlv, 51000.nlv. (former files: Maimp.d, Meimp.d, Nimpotr.d,nyimp.d, P94imp.d, vtimp.d).

Batch file to Update County database with corrected .nlv External files.

# Update County database with corrected .nlv External files.

#

copy T:\NMIMRuns\MSAT\NLEVFiles\Maimp.d C:\mysql\data\NCD20060725\ExternalFiles\25000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\Meimp.d C:\mysql\data\NCD20060725\ExternalFiles\23000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\vtimp.d C:\mysql\data\NCD20060725\ExternalFiles\50000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nyimp.d C:\mysql\data\NCD20060725\ExternalFiles\36000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\09000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\10000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\11000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\24000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\33000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\34000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\42000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\44000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060725\ExternalFiles\51000.nlv /a /y ;

# Update the STATE table where these external files are referenced:

use NCD20060725;

update state set NLEVFileName='34000.nlv' where fipsstateid=34;

update state set NLEVFileName='33000.nlv' where fipsstateid=33;

update state set NLEVFileName='09000.nlv' where fipsstateid=9;

update state set NLEVFileName='11000.nlv' where fipsstateid=11;

**2.** The SCCToxics table has evaporative ratios for Dies, LPG, and CNG, and it shouldn't.

Script to set all evap emission factors in the SCCToxics table to NULL for all cases where fueltype is dies, lpg, and cng.:

update scctoxics t, scc s

set

t.evapEthGas = NULL,

t.evapBaseGas = NULL,

t.evapMTBEGas = NULL,

t.evapRFGGas = NULL

where s.fuelType in ('Dies','LPG','CNG') and s.sccid = t.sccid;

Script to set the RatioType to PM10 for naphthalene for all diesel SCCs. (It was incorrectly set to PMVOC for onroad diesel).

update scctoxics t, scc s

set t.ratioType='PM10'

where t.pollutantCodeId=52

and s.fuelType='Dies' and t.sccid=s.sccid;

# Change Log for NCD20060612

The PM25fac field in the SCC table has been changed from 0.92 to 0.97 for diesel nonroad equipment. (PM25fac for nonroad Gasoline, CNG, and LPG equipment remains unchanged at 0.92, 1.00, and 1.00, respectively, and NULL for all onroad vehicles.) The PM25fac field is used to calculate nonroad PM2.5 as a fraction of PM10: PM2.5 = PM25fac x PM10.

The correct diesel PM25fac field (0.97) and correct mgNH3perGallon (diesel=83.3, gasoline=116) were first instituted in County20040229, and remained correct in County20040325, and County20040412. However, both reverted to their prior, incorrect values for mgNH3perGallon (diesel=165.86, gasoline=153.47) and for diesel PM25fac (0.92) in County20040508. The mgNH3perGallon values were corrected in NCD20050714, but the erroneous diesel PM25fac value was not discovered until this database.

The MySQL script implementing this change is as follows:

# FixPM25.sql

# Script to fix PM25fac in the SCC table

# First copy NCD20060201 into NC20060612

use NCD20060612;

update scc

set PM25fac=0.97

where FuelType='Dies'

and ORNR='NR'

;

The NONROAD technical document which describes the basis for PM25fac is “Exhaust and Crankcase Emission Factors for Nonroad Engine Modeling -- Compression-Ignition," NR-009c (EPA420-P-04-009, April 2004),” which may be found at http://www.epa.gov/otaq/models/nonrdmdl/nonrdmdl2004/420p04009.pdf..

# Change Log for NCD20060201

Table countyyear updated to set stage2pct=0 for all states.

ExternalFiles: File 19000.sea replaced with new version. (The old version had only a few SCCs, and missing SCCs are given a flat seasonal allocation, resulting, among other things, in snow equipment emissions in the summer.)

The following line was added to External File 08000oil.alo:

OIL 08014 1996 0 Broomfield County

The choice of zero for the indicator value matches Boulder County and is also the value assigned in the NONROAD2005 default file, CO\_OIL.ALO. (However, it should be noted that the value for Boulder County in this CO\_OIL.ALO is 46.)

# Change Log for NCD20051207

Changes were submitted by states in terms of which counties were grouped together for use of gasoline and diesel fuels for 2002. If a county was changed from one grouping to another, this change was also made for all other calendar years, both forward (2003 and later calendar years) and backward (1999, 2000 and 2001 calendar years). When a county is determined to use the same gasoline as used by another county, we have assumed that all other fuel parameters are switched to the new county as well. This should not change anything inappropriately, since gasoline parameters are normally more detailed than for the other fuels.

# Colorado

# Set counties 13 & 14 equal to county 1 in

# all calendar years and months.

#

delete from ncd20051207.countyyearmonth

where fipsstateid=8 and fipscountyid in (13,14);

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select 13,

FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource

from ncd20050928.countyyearmonth

where fipsstateid=8 and fipscountyid=1;

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select 14,

FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource

from ncd20050928.countyyearmonth

where fipsstateid=8 and fipscountyid=1;

# Delaware

# Set all counties month=10 equal to

# month=11 in all calendar years.

#

delete from ncd20051207.countyyearmonth

where fipsstateid=10 and fipscountyid in (1,3,5) and

month=10;

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select FIPSCountyId,FIPSStateId,Year,

10,

NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

1006,1006

from ncd20050928.countyyearmonth

where fipsstateid=10 and fipscountyid in (1,3,5) and

Month=11;

# Maryland

# Change highway diesel sulfur values

# for calendar years up to controls.

#

# Use 2002 sulfur values for 1999 through 2006.

# Set June through December 2006 to 43 ppm.

#

delete from ncd20051207.countyyearmonth

where fipsstateid=24;

#

# Calendar years with changes.

#

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select a.FIPSCountyId,a.FIPSStateId,a.Year,a.Month,a.NGId,

b.HwyDieselId,

a.HwyGasolineId,a.NRGasolineId,a.NRDieselId,a.RMDieselId,

b.HwyFuelDataSource,

a.NRFuelDataSource

from ncd20050928.countyyearmonth as a,

ncd20050928.countyyearmonth as b

where a.fipsstateid=b.fipsstateid and

a.fipscountyid=b.fipscountyid and

a.month=b.month and

a.fipsstateid=24 and

a.year in (1999,2000,2001,2003,2004,2005,2006) and

b.year=2002;

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select FIPSCountyId,FIPSStateId,Year,Month,NGId,

HwyDieselId,HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource

from ncd20050928.countyyearmonth

where fipsstateid=24 and year=2002;

#

# All other calendar years.

#

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select FIPSCountyId,FIPSStateId,Year,Month,NGId,

HwyDieselId,HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource

from ncd20050928.countyyearmonth

where fipsstateid=24 and year>2006;

#

# June 2006 diesel sulfur value of 43 ppm.

#

Update ncd20051207.countyyearmonth

set hwydieselid=43

where fipsstateid=24 and year=2006 and month>5;

# Michigan

# Set counties 93 & 161 equal to county 87 in

# all calendar years and months.

#

delete from ncd20051207.countyyearmonth

where fipsstateid=26 and fipscountyid in (93,161);

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select 93,

FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource

from ncd20050928.countyyearmonth

where fipsstateid=26 and fipscountyid=87;

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select 161,

FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource

from ncd20050928.countyyearmonth

where fipsstateid=26 and fipscountyid=87;

# Minnesota

# Set all fuel information for all counties

# in all years equal to the fuels in countyid=3.

#

delete from ncd20051207.countyyearmonth

where fipsstateid=27;

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select b.FIPSCountyId,

a.FIPSStateId,a.Year,a.Month,a.NGId,

a.HwyDieselId,a.HwyGasolineId,a.NRGasolineId,a.NRDieselId,

a.RMDieselId,a.HwyFuelDataSource,a.NRFuelDataSource

from ncd20050928.countyyearmonth as a,

ncd20050928.county as b

where a.fipsstateid=b.fipsstateid and

a.fipsstateid=27 and

a.fipscountyid=3;

# Virginia

# Set counties 36 equal to county 1 in

# all calendar years and months.

#

delete from ncd20051207.countyyearmonth

where fipsstateid=51 and fipscountyid=36;

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select 36,

FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource

from ncd20050928.countyyearmonth

where fipsstateid=51 and fipscountyid=1;

# Washington

# Set all fuels to one of two possible

# county groupings.

# Set Spokane County winter gasoline to be

# the same as for Missoula County, Montana.

#

#

# Calender years 2004 and later.

#

delete from ncd20051207.countyyearmonth

where fipsstateid=53 and year>=2003;

#Query OK, 21996 rows affected (0.44 sec)

drop table if exists ncd20051207.wagas;

create table ncd20051207.wagas

select distinct fipsstateid,fipscountyid,hwygasolineid

from ncd20050928.countyyearmonth

where fipsstateid=53 and

hwygasolineid=159;

#Query OK, 20 rows affected (0.36 sec)

#Records: 20 Duplicates: 0 Warnings: 0

insert into ncd20051207.wagas

(fipsstateid,fipscountyid,hwygasolineid)

select distinct fipsstateid,fipscountyid,hwygasolineid

from ncd20050928.countyyearmonth

where fipsstateid=53 and

hwygasolineid=295;

#Query OK, 19 rows affected (0.30 sec)

#Records: 19 Duplicates: 0 Warnings: 0

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select b.FIPSCountyId,

a.FIPSStateId,a.Year,a.Month,a.NGId,

a.HwyDieselId,a.HwyGasolineId,a.NRGasolineId,a.NRDieselId,

a.RMDieselId,a.HwyFuelDataSource,a.NRFuelDataSource

from ncd20050928.countyyearmonth as a,

ncd20051207.wagas as b

where a.fipsstateid=b.fipsstateid and

a.fipsstateid=53 and

a.year>=2003 and

a.fipscountyid=1 and

b.hwygasolineid=159;

#Query OK, 11280 rows affected (1.14 sec)

#Records: 11280 Duplicates: 0 Warnings: 0

insert into ncd20051207.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,

HwyGasolineId,NRGasolineId,NRDieselId,RMDieselId,

HwyFuelDataSource,NRFuelDataSource)

select b.FIPSCountyId,

a.FIPSStateId,a.Year,a.Month,a.NGId,

a.HwyDieselId,a.HwyGasolineId,a.NRGasolineId,a.NRDieselId,

a.RMDieselId,a.HwyFuelDataSource,a.NRFuelDataSource

from ncd20050928.countyyearmonth as a,

ncd20051207.wagas as b

where a.fipsstateid=b.fipsstateid and

a.fipsstateid=53 and

a.year>=2003 and

a.fipscountyid=9 and

b.hwygasolineid=295;

#Query OK, 10716 rows affected (0.28 sec)

#Records: 10716 Duplicates: 0 Warnings: 0

drop table if exists ncd20051207.wagas;

#

# Special winter fuels for Spokane County (53063)

# Same as Missoula County, Montana (30063)

#

update ncd20051207.countyyearmonth

set hwygasolineid=338,

nrgasolineid=338

where fipsstateid=53 and

fipscountyid=63 and

month in (1,2,9,10,11,12) and

year in (1999,2000,2001,2002,2003);

#Query OK, 0 rows affected (0.00 sec)

#Rows matched: 30 Changed: 0 Warnings: 0

update ncd20051207.countyyearmonth

set hwygasolineid=673,

nrgasolineid=673

where fipsstateid=53 and

fipscountyid=63 and

month in (1,2,9,10,11,12) and

year in (2004,2005);

#Query OK, 12 rows affected (0.00 sec)

#Rows matched: 12 Changed: 12 Warnings: 0

update ncd20051207.countyyearmonth

set hwygasolineid=1828,

nrgasolineid=1828

where fipsstateid=53 and

fipscountyid=63 and

month in (1,2,9,10,11,12) and

year in (2006);

#Query OK, 6 rows affected (0.00 sec)

#Rows matched: 6 Changed: 6 Warnings: 0

update ncd20051207.countyyearmonth

set hwygasolineid=1831,

nrgasolineid=1831

where fipsstateid=53 and

fipscountyid=63 and

month in (1,2,9,10,11,12) and

year in (2007);

#Query OK, 6 rows affected (0.00 sec)

#Rows matched: 6 Changed: 6 Warnings: 0

update ncd20051207.countyyearmonth

set hwygasolineid=1834,

nrgasolineid=1834

where fipsstateid=53 and

fipscountyid=63 and

month in (1,2,9,10,11,12) and

year>2007;

#Query OK, 258 rows affected (0.00 sec)

#Rows matched: 258 Changed: 258 Warnings: 0

The Stage2Pct field of the CountyYear table affects the vapor displacement pump-based refueling emissions from gasoline nonroad sources. This field is included in the options file generated by NMIM for the NONROAD model. Some states have provided values for the Stage2Pct.

select fipsstateid,fipscountyid,stage2pct,year

from countyyear where stage2pct>0 order by fipsstateid;

+‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑+

| fipsstateid | fipscountyid | stage2pct | year |

+‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑+

| 11 | 1 | 90.0 | 2002 |

| 24 | 15 | 100.0 | 2002 |

| 24 | 17 | 100.0 | 2002 |

| 24 | 21 | 100.0 | 2002 |

| 24 | 9 | 100.0 | 2002 |

| 24 | 510 | 100.0 | 2002 |

| 24 | 25 | 100.0 | 2002 |

| 24 | 3 | 100.0 | 2002 |

| 24 | 27 | 100.0 | 2002 |

| 24 | 5 | 100.0 | 2002 |

| 24 | 31 | 100.0 | 2002 |

| 24 | 33 | 100.0 | 2002 |

| 24 | 13 | 100.0 | 2002 |

| 24 | 11 | 100.0 | 2002 |

| 51 | 510 | 77.0 | 2002 |

| 51 | 600 | 77.0 | 2002 |

| 51 | 610 | 77.0 | 2002 |

| 51 | 13 | 77.0 | 2002 |

| 51 | 683 | 77.0 | 2002 |

| 51 | 59 | 77.0 | 2002 |

| 51 | 685 | 77.0 | 2002 |

| 51 | 107 | 77.0 | 2002 |

| 51 | 153 | 77.0 | 2002 |

| 51 | 179 | 77.0 | 2002 |

| 51 | 87 | 77.0 | 2002 |

| 51 | 670 | 77.0 | 2002 |

| 51 | 760 | 77.0 | 2002 |

| 51 | 85 | 77.0 | 2002 |

| 51 | 41 | 77.0 | 2002 |

| 51 | 570 | 77.0 | 2002 |

+‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑+

30 rows in set (0.06 sec)

The fraction of fuel pumped (as opposed to dispensed from gas cans) for nonroad sources is considered very uncertain. EPA also considers the effectiveness of currently available Stage 2 technology when applied to nonroad sources to be uncertain. For purposes of NMIM at this time, all Stage 2 controls, as applied to nonroad sources, will be assumed to have no (zero) effectiveness. This means that all existing data, the values supplied by states for the Stage2Pct field will be set to zero.

update ncd20051207.countyyear

set stage2pct=0.0;

In the CountyYear table, 2002 ATPFileName values were copied to all 2003 and later calendar years. If there was no value, the 2002 ATPFileName values were applied to the 1999 through 2001 calendar years as well. All of the other 2002 file names, other than diesel sales fractions and I/M program descriptions, were applied to all calendar years. If changes were made, the DataSourceId was updated to be the same as for 2002.

drop table if exists countyyear2002;

create table countyyear2002

select a.FIPSCountyId,

a.FIPSStateId,

a.Year,

a.ATPFileName,

a.IMFileName,

a.RegDistFileName,

a.VMTByHourFileName,

a.MileAccumFileName,

a.TripsPerDayFileName,

a.StartDistFileName,

a.SoakDistFileName,

a.HotSoakFileName,

a.DiurnSoakActivityFileName,

a.TripLengthFileName,

a.DieselFractFileName,

a.AvgSpeedDistBaseFileName,

a.Stage2Pct,

a.NRACTFileName,

a.DataSourceId,

a.HasNRStateInputs,

a.HasORStateInputs,

b.ATPFileName as ATP,

b.RegDistFileName as RegDist,

b.VMTByHourFileName as VMTByHour,

b.MileAccumFileName as MileAccum,

b.TripsPerDayFileName as TripsPerDay,

b.StartDistFileName as StartDist,

b.SoakDistFileName as SoakDist,

b.HotSoakFileName as HotSoak,

b.DiurnSoakActivityFileName as DiurnSoakActivity,

b.TripLengthFileName as TripLength,

b.AvgSpeedDistBaseFileName as AvgSpeedDistBase,

b.NRACTFileName as NRACT,

b.DataSourceId as DataSource

from ncd20050928.countyyear as a,

ncd20050928.countyyear as b

where a.fipsstateid=b.fipsstateid and

a.fipscountyid=b.fipscountyid and

b.year=2002;

update countyyear2002

set atpfilename=atp

where atpfilename="";

update countyyear2002

set atpfilename=atp

where atp="" and

atpfilename<>"" and

year>2002;

update countyyear2002

set regdistfilename=regdist

where regdistfilename="";

update countyyear2002

set vmtbyhourfilename=vmtbyhour

where vmtbyhourfilename="";

update countyyear2002

set mileaccumfilename=mileaccum

where mileaccumfilename="";

update countyyear2002

set tripsperdayfilename=tripsperday

where tripsperdayfilename="";

update countyyear2002

set startdistfilename=startdist

where startdistfilename="";

update countyyear2002

set soakdistfilename=soakdist

where soakdistfilename="";

update countyyear2002

set hotsoakfilename=hotsoak

where hotsoakfilename="";

update countyyear2002

set diurnsoakactivityfilename=diurnsoakactivity

where diurnsoakactivityfilename="";

update countyyear2002

set triplengthfilename=triplength

where triplengthfilename="";

update countyyear2002

set avgspeeddistbasefilename=avgspeeddistbase

where avgspeeddistbasefilename="";

update countyyear2002

set nractfilename=nract

where nractfilename="";

update countyyear2002

set datasourceid=datasource

where atp<>"";

update countyyear2002

set datasourceid=datasource

where regdist<>"";

update countyyear2002

set datasourceid=datasource

where vmtbyhour<>"";

update countyyear2002

set datasourceid=datasource

where mileaccum<>"";

update countyyear2002

set datasourceid=datasource

where tripsperday<>"";

update countyyear2002

set datasourceid=datasource

where startdist<>"";

update countyyear2002

set datasourceid=datasource

where soakdist<>"";

update countyyear2002

set datasourceid=datasource

where hotsoak<>"";

update countyyear2002

set datasourceid=datasource

where diurnsoakactivity<>"";

update countyyear2002

set datasourceid=datasource

where triplength<>"";

update countyyear2002

set datasourceid=datasource

where avgspeeddistbase<>"";

update countyyear2002

set datasourceid=datasource

where nract<>"";

truncate table ncd20051207.countyyear;

insert into ncd20051207.countyyear

(FIPSCountyId,FIPSStateId,Year,ATPFileName,IMFileName,

RegDistFileName,VMTByHourFileName,MileAccumFileName,

TripsPerDayFileName,StartDistFileName,SoakDistFileName,

HotSoakFileName,DiurnSoakActivityFileName,

TripLengthFileName,DieselFractFileName,

AvgSpeedDistBaseFileName,Stage2Pct,

NRACTFileName,DataSourceId,HasNRStateInputs,

HasORStateInputs)

select FIPSCountyId,FIPSStateId,Year,ATPFileName,IMFileName,

RegDistFileName,VMTByHourFileName,MileAccumFileName,

TripsPerDayFileName,StartDistFileName,SoakDistFileName,

HotSoakFileName,DiurnSoakActivityFileName,

TripLengthFileName,DieselFractFileName,

AvgSpeedDistBaseFileName,Stage2Pct,

NRACTFileName,DataSourceId,HasNRStateInputs,

HasORStateInputs

from countyyear2002;

drop table countyyear2002;

External data files were created with calendar specific diesel sales fractions based on the state supplied fractions for the 2002 calendar year. There is a separate external data file for each calendar year for each county which provided diesel sales fractions. These file names were added to the CountyYear table.

drop table if exists dieselsf;

CREATE TABLE `dieselsf` (

`FIPSStateId` smallint NOT NULL default '0',

`FIPSCountyId` smallint NOT NULL default '0',

`year` smallint(6) default NULL,

`dieselFracFileName` char(7) NOT NULL default '',

PRIMARY KEY (`FIPSStateId`,`FIPSCountyId`,`year`)

) TYPE=MyISAM;

truncate table dieselsf;

load data infile 'e:/nmimdata/database/retrofit/dffnallyears.txt' replace

into table dieselsf

fields terminated by '\t'

optionally enclosed by '"'

escaped by '\\'

lines terminated by '\n'

ignore 1 lines

(fipsstateid,fipscountyid,year,dieselfracfilename);

CREATE TABLE ncd20051207.countyyeard (

`FIPSCountyId` smallint(6) NOT NULL default '0',

`FIPSStateId` tinyint(2) NOT NULL default '0',

`Year` smallint(6) NOT NULL default '0',

`ATPFileName` char(12) NOT NULL default '',

`IMFileName` char(12) NOT NULL default '',

`RegDistFileName` char(8) NOT NULL default '',

`VMTByHourFileName` char(8) NOT NULL default '',

`MileAccumFileName` char(8) NOT NULL default '',

`TripsPerDayFileName` char(8) NOT NULL default '',

`StartDistFileName` char(8) NOT NULL default '',

`SoakDistFileName` char(8) NOT NULL default '',

`HotSoakFileName` char(8) NOT NULL default '',

`DiurnSoakActivityFileName` char(8) NOT NULL default '',

`TripLengthFileName` char(8) NOT NULL default '',

`DieselFractFileName` char(8) NOT NULL default '',

`AvgSpeedDistBaseFileName` char(8) NOT NULL default '',

`Stage2Pct` decimal(4,1) default NULL,

`NRACTFileName` char(8) NOT NULL default '',

`DataSourceId` smallint(6) default NULL,

`HasNRStateInputs` char(1) NOT NULL default 'N',

`HasORStateInputs` char(1) NOT NULL default 'N',

PRIMARY KEY (`FIPSCountyId`,`FIPSStateId`,`Year`)

) TYPE=MyISAM ;

truncate table ncd20051207.countyyeard;

insert into ncd20051207.countyyeard

(FIPSCountyId,FIPSStateId,Year,ATPFileName,IMFileName,

RegDistFileName,VMTByHourFileName,MileAccumFileName,

TripsPerDayFileName,StartDistFileName,SoakDistFileName,

HotSoakFileName,DiurnSoakActivityFileName,

TripLengthFileName,DieselFractFileName,

AvgSpeedDistBaseFileName,Stage2Pct,

NRACTFileName,DataSourceId,HasNRStateInputs,

HasORStateInputs)

select a.FIPSCountyId,

a.FIPSStateId,

a.Year,

a.ATPFileName,

a.IMFileName,

a.RegDistFileName,

a.VMTByHourFileName,

a.MileAccumFileName,

a.TripsPerDayFileName,

a.StartDistFileName,

a.SoakDistFileName,

a.HotSoakFileName,

a.DiurnSoakActivityFileName,

a.TripLengthFileName,

b.DieselFracFileName,

a.AvgSpeedDistBaseFileName,

a.Stage2Pct,

a.NRACTFileName,

a.DataSourceId,

a.HasNRStateInputs,

a.HasORStateInputs

from ncd20051207.countyyear as a left join

dieselsf as b using (fipsstateid,fipscountyid,year);

drop table if exists ncd20051207.countyyear;

rename table ncd20051207.countyyeard to ncd20051207.countyyear;

A script was used to change the database to better utilize the data source information in the database. Some new fields were created to store data source information and other fields were moved to reduce duplication.

#

# Intended to fix the problems identified in the data source fields in

# the NCD20050318 version of the database.

#

use ncd20051207;

delete from datasource where datasourceid=701;

insert ignore into datasource

(datasourceid,author,date,sponsor,documentid,qualitylevel,comments)

values ("701","Pechan","2004‑01‑01","EPA","nei\_onroad\_jan04.pdf",NULL," ftp://ftp.epa.gov/EmisInventory/finalnei99ver3/haps/documentation/onroad/nei\_onroad\_jan04.pdf");

# Add datasourceid #624 to the data source table to explain Broomfield County (CO)

# in CountyMonthHour.

# Change datasourceid to 624 in other tables where Broomfied County was added.

insert ignore into datasource

(datasourceid,author,date,sponsor,documentid,qualitylevel,comments)

values ("624","OTAQ","2004‑12‑01","EPA","Same as Boulder: 310,407,605 & 623.",NULL,

"Broomfield assumes same data as Boulder. Refer to data sources 310,407,605 & 623.");

update countymonthhour set datasourceid=624 where fipsstateid=8 and fipscountyid=14;

update countyyear set datasourceid=624 where fipsstateid=8 and fipscountyid=14;

update countyyearmonth set hwyfueldatasource=624

where fipsstateid=8 and fipscountyid=14 and year=2002;

update countyyearmonth set nrfueldatasource=624

where fipsstateid=8 and fipscountyid=14 and year=2002;

# Replace NULL values for District of Columbia (11) for 2002 in BaseYearVMT.

update baseyearvmt set datasourceid=1101 where fipsstateid=11 and baseyear=2002 and vmt=0;

# Add data source #804 to the data source table to explain BaseYearVMT for 1999.

# Replace NULL values for 1999 VMT in BaseYearVMT.

insert ignore into datasource

(datasourceid,author,date,sponsor,documentid,qualitylevel,comments)

values ("804","Pechan","2003‑09‑12","EPA","VMT99\_f\_m6\_nei3.dbf",NULL,

"This is the final,version 3, of the 1999 NEI updates to the VMT.");

update baseyearvmt set datasourceid=804 where baseyear=1999;

#

# Make structural changes to the database tables.

# Move data source information to the appropriate new fields in other tables.

#

# Save the old County table and create a new version.

drop table if exists oldcounty;

alter table county rename oldcounty;

CREATE TABLE `newcounty` (

`FIPSCountyId` smallint(6) NOT NULL default '0',

`FIPSStateId` tinyint(2) NOT NULL default '0',

`CountyName` char(50) NOT NULL default '',

`Altitude` char(1) NOT NULL default '',

`BarometricPressure` float NOT NULL default '0',

`OzoneSeasonStartMonth` smallint(6) NOT NULL default '0',

`OzoneSeasonStartDay` smallint(6) NOT NULL default '0',

`OzoneSeasonEndMonth` smallint(6) NOT NULL default '0',

`OzoneSeasonEndDay` smallint(6) NOT NULL default '0',

`Stage2StartYear` smallint(6) default NULL,

`PhaseInYears` smallint(6) default NULL,

`LDVStage2Percent` float default NULL,

`HDVStage2Percent` float default NULL,

`NGVFractionFileName` char(8) NOT NULL default '',

`DataSourceId` smallint(6) NOT NULL default '0',

`CountyVMTMonthAllocationDataSource` smallint(6) NOT NULL default '0',

`CountyMonthHourDataSource` smallint(6) NOT NULL default '0',

PRIMARY KEY (`FIPSCountyId`,`FIPSStateId`)

) TYPE=MyISAM ;

# Add in the data sources from the CountyMonthHour table.

# Set the County table data sources to the default (1999 NEI) data source.

truncate newcounty;

insert into newcounty

(fipscountyid,fipsstateid,countyname,altitude,barometricpressure,

ozoneseasonstartmonth,ozoneseasonstartday,ozoneseasonendmonth,ozoneseasonendday,

stage2startyear,phaseinyears,ldvstage2percent,hdvstage2percent,

ngvfractionfilename,datasourceid,countyvmtmonthallocationdatasource,

countymonthhourdatasource)

select a.fipscountyid,a.fipsstateid,countyname,altitude,barometricpressure,

ozoneseasonstartmonth,ozoneseasonstartday,ozoneseasonendmonth,ozoneseasonendday,

stage2startyear,phaseinyears,ldvstage2percent,hdvstage2percent,

ngvfractionfilename,701,0,b.datasourceid

from oldcounty as a,countymonthhour as b

where a.fipsstateid=b.fipsstateid and

a.fipscountyid=b.fipscountyid and

b.month=1 and

b.hourid=1;

# Create the final County table.

CREATE TABLE `county` (

`FIPSCountyId` smallint(6) NOT NULL default '0',

`FIPSStateId` tinyint(2) NOT NULL default '0',

`CountyName` char(50) NOT NULL default '',

`Altitude` char(1) NOT NULL default '',

`BarometricPressure` float NOT NULL default '0',

`OzoneSeasonStartMonth` smallint(6) NOT NULL default '0',

`OzoneSeasonStartDay` smallint(6) NOT NULL default '0',

`OzoneSeasonEndMonth` smallint(6) NOT NULL default '0',

`OzoneSeasonEndDay` smallint(6) NOT NULL default '0',

`Stage2StartYear` smallint(6) default NULL,

`PhaseInYears` smallint(6) default NULL,

`LDVStage2Percent` float default NULL,

`HDVStage2Percent` float default NULL,

`NGVFractionFileName` char(8) NOT NULL default '',

`DataSourceId` smallint(6) NOT NULL default '0',

`CountyVMTMonthAllocationDataSource` smallint(6) NOT NULL default '0',

`CountyMonthHourDataSource` smallint(6) NOT NULL default '0',

PRIMARY KEY (`FIPSCountyId`,`FIPSStateId`)

) TYPE=MyISAM ;

# Add in the data sources from the CountyVMTMonthAllocation table.

truncate county;

insert into county

(fipscountyid,fipsstateid,countyname,altitude,barometricpressure,

ozoneseasonstartmonth,ozoneseasonstartday,ozoneseasonendmonth,ozoneseasonendday,

stage2startyear,phaseinyears,ldvstage2percent,hdvstage2percent,

ngvfractionfilename,datasourceid,countyvmtmonthallocationdatasource,

countymonthhourdatasource)

select a.fipscountyid,a.fipsstateid,countyname,altitude,barometricpressure,

ozoneseasonstartmonth,ozoneseasonstartday,ozoneseasonendmonth,ozoneseasonendday,

stage2startyear,phaseinyears,ldvstage2percent,hdvstage2percent,

ngvfractionfilename,a.datasourceid,b.datasourceid,countymonthhourdatasource

from newcounty as a left join countyvmtmonthallocation as b

on a.fipsstateid=b.fipsstateid and

a.fipscountyid=b.fipscountyid

where (b.vtype=1 and b.roadtype=11 and b.month=1) or isnull(b.datasourceid);

# Save the old version of the CountyYear table and create a new one.

drop table if exists oldcountyyear;

alter table countyyear rename oldcountyyear;

CREATE TABLE `countyyear` (

`FIPSCountyId` smallint(6) NOT NULL default '0',

`FIPSStateId` tinyint(2) NOT NULL default '0',

`Year` smallint(6) NOT NULL default '0',

`ATPFileName` char(12) NOT NULL default '',

`IMFileName` char(12) NOT NULL default '',

`RegDistFileName` char(8) NOT NULL default '',

`VMTByHourFileName` char(8) NOT NULL default '',

`MileAccumFileName` char(8) NOT NULL default '',

`TripsPerDayFileName` char(8) NOT NULL default '',

`StartDistFileName` char(8) NOT NULL default '',

`SoakDistFileName` char(8) NOT NULL default '',

`HotSoakFileName` char(8) NOT NULL default '',

`DiurnSoakActivityFileName` char(8) NOT NULL default '',

`TripLengthFileName` char(8) NOT NULL default '',

`DieselFractFileName` char(8) NOT NULL default '',

`AvgSpeedDistBaseFileName` char(8) NOT NULL default '',

`Stage2Pct` decimal(4,1) default NULL,

`NRACTFileName` char(8) NOT NULL default '',

`DataSourceId` smallint(6) default NULL,

`HasNRStateInputs` char(1) NOT NULL default 'N',

`HasORStateInputs` char(1) NOT NULL default 'N',

`BaseYearVMTDataSource` smallint(6) NOT NULL default '0',

`CountyYearMonthHourDataSource` smallint(6) NOT NULL default '0',

PRIMARY KEY (`FIPSCountyId`,`FIPSStateId`,`Year`)

) TYPE=MyISAM ;

# Create a smaller version of the BaseYearVMT table.

drop table if exists yearvmt;

create table YearVMT

select fipsstateid,fipscountyid,baseyear as year,datasourceid

from baseyearvmt

group by fipsstateid,fipscountyid,baseyear;

alter table yearvmt add primary key (fipsstateid,fipscountyid,year);

# Add in the data sources from the smaller BaseYearVMT table.

truncate countyyear;

insert into countyyear

(fipscountyid,fipsstateid,year,atpfilename,imfilename,regdistfilename,vmtbyhourfilename,

mileaccumfilename,tripsperdayfilename,startdistfilename,soakdistfilename,

hotsoakfilename,diurnsoakactivityfilename,triplengthfilename,dieselfractfilename,

avgspeeddistbasefilename,stage2pct,nractfilename,datasourceid,hasnrstateinputs,

hasorstateinputs,baseyearvmtdatasource,countyyearmonthhourdatasource)

select a.fipscountyid,a.fipsstateid,a.year,atpfilename,imfilename,regdistfilename,

vmtbyhourfilename,

mileaccumfilename,tripsperdayfilename,startdistfilename,soakdistfilename,

hotsoakfilename,diurnsoakactivityfilename,triplengthfilename,dieselfractfilename,

avgspeeddistbasefilename,stage2pct,nractfilename,a.datasourceid,hasnrstateinputs,

hasorstateinputs,b.datasourceid,0

from oldcountyyear as a left join yearvmt as b

on a.fipsstateid=b.fipsstateid and

a.fipscountyid=b.fipscountyid and

a.year=b.year;

# Set the CountyYearMonthHour table data sources to the default (AIR) data sources.

update countyyear set countyyearmonthhourdatasource=612 where year=1999;

update countyyear set countyyearmonthhourdatasource=613 where year=2000;

update countyyear set countyyearmonthhourdatasource=616 where year=2001;

update countyyear set countyyearmonthhourdatasource=617 where year=2002;

update countyyear set countyyearmonthhourdatasource=618 where year=2003;

#

# Remove unneeded data source fields.

#

alter table countyvmtmonthallocation drop datasourceid;

alter table countymonthhour drop datasourceid;

alter table baseyearvmt drop datasourceid;

#

# Drop unneeded working tables.

#

drop table oldcounty;

drop table newcounty;

drop table oldcountyyear;

drop table yearvmt;

The diesel sales fraction data files provided by states for the 2002 calendar year were used to generate state specific diesel sales fractions for all calendar years. This will allow modeling of calendar years other than 2002, but still use state specific diesel sales fractions, rather than the default values.

drop table if exists dieselsf;

CREATE TABLE `dieselsf` (

`FIPSStateId` smallint NOT NULL default '0',

`FIPSCountyId` smallint NOT NULL default '0',

`year` smallint(6) default NULL,

`dieselFracFileName` char(7) NOT NULL default '',

PRIMARY KEY (`FIPSStateId`,`FIPSCountyId`,`year`)

) TYPE=MyISAM;

truncate table dieselsf;

load data infile 'e:/nmimdata/database/retrofit/dffnallyearsnew.txt' replace

into table dieselsf

fields terminated by '\t'

optionally enclosed by '"'

escaped by '\\'

lines terminated by '\n'

ignore 1 lines

(fipsstateid,fipscountyid,year,dieselfracfilename);

update ncd20051207.countyyear as a, dieselsf as b

set a.dieselfractfilename=b.dieselfracfilename

where a.fipsstateid=b.fipsstateid and

a.fipscountyid=b.fipscountyid and

a.year=b.year;

In order that the fleet option of NMIM can be used for all calendar years, VMT values for 2050 were added to allow for interpolation between 2002 and 2050. The values used for 2050 are identical to those for the 2002 calendar year.

delete from ncd20051207.baseyearvmt

where baseyear=2050;

insert into ncd20051207.baseyearvmt

(BaseYear,

VClass,

RoadType,

FIPSCountyId,

FIPSStateId,

VMT)

select 2050,

VClass,

RoadType,

FIPSCountyId,

FIPSStateId,

VMT

from ncd20050928.baseyearvmt

where baseyear=2002;

Broomfield County, Colorado (08014) was created in calendar year 2001. Since there is no VMT for this county in 2001, it cannot be properly modeled until calendar year 2002. All references to Broomfield before 2002 are removed and Broomfield County is set equal to Boulder County for 2002 and all future calendar years in the CountyYearMonth table.

delete from ncd20051207.countyyearmonth

where fipsstateid=8 and

fipscountyid=14 and

year in (1999,2000,2001);

insert into ncd20051207.countyyear

(FIPSCountyId,FIPSStateId,Year,ATPFileName,IMFileName,

RegDistFileName,VMTByHourFileName,MileAccumFileName,

TripsPerDayFileName,StartDistFileName,SoakDistFileName,

HotSoakFileName,DiurnSoakActivityFileName,

TripLengthFileName,DieselFractFileName,

AvgSpeedDistBaseFileName,Stage2Pct,

NRACTFileName,DataSourceId,HasNRStateInputs,

HasORStateInputs)

select 14,

FIPSStateId,Year,ATPFileName,IMFileName,

RegDistFileName,VMTByHourFileName,MileAccumFileName,

TripsPerDayFileName,StartDistFileName,SoakDistFileName,

HotSoakFileName,DiurnSoakActivityFileName,

TripLengthFileName,DieselFractFileName,

AvgSpeedDistBaseFileName,Stage2Pct,

NRACTFileName,DataSourceId,HasNRStateInputs,

HasORStateInputs

from ncd20050928.countyyear

where fipsstateid=8 and

fipscountyid=13 and

year>2002;

Changes were made in a variety of fields in several tables to better store the data. These changes are also necessary to keep the data tables consistent with the documentation for the database in the Erwin application representation of this database.

alter table ncd20051207.County modify OzoneSeasonStartMonth TINYINT(2) NOT NULL DEFAULT '0' ;

alter table ncd20051207.County modify OzoneSeasonStartDay TINYINT(2) NOT NULL DEFAULT '0' ;

alter table ncd20051207.County modify OzoneSeasonEndMonth TINYINT(2) NOT NULL DEFAULT '0' ;

alter table ncd20051207.County modify OzoneSeasonEndDay TINYINT(2) NOT NULL DEFAULT '0' ;

alter table ncd20051207.CountyYear modify Stage2Pct FLOAT DEFAULT NULL ;

alter table ncd20051207.CountyYearMonth modify FIPSStateId TINYINT(2) NOT NULL DEFAULT '0' ;

alter table ncd20051207.CountyYearMonth modify Month TINYINT(2) NOT NULL DEFAULT '0' ;

alter table ncd20051207.CountyYearMonth modify HwyDieselId SMALLINT(6) DEFAULT NULL ;

alter table ncd20051207.Diesel modify DieselSulfur FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify RVP FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify GasSulfur FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify GasMaxSulfur FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify RVPOxyWaiver TINYINT(1) NOT NULL DEFAULT '0' ;

alter table ncd20051207.Gasoline modify ETOHVolume FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify ETOHMktShare FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify MTBEVolume FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify MTBEMktShare FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify ETBEVolume FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify ETBEMktShare FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify TAMEVolume FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify TAMEMktShare FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify AromaticContent FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify OlefinContent FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify BenzeneContent FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify E200 FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.Gasoline modify E300 FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.NaturalGas modify NGSulfur FLOAT NOT NULL DEFAULT '0.00' ;

alter table ncd20051207.SCC modify mgNH3perGallon FLOAT DEFAULT NULL ;

alter table ncd20051207.SCC modify TOGfac FLOAT DEFAULT NULL ;

alter table ncd20051207.SCC modify NMOGfac FLOAT DEFAULT NULL ;

alter table ncd20051207.SCC modify NMHCfac FLOAT DEFAULT NULL ;

alter table ncd20051207.SCC modify VOCfac FLOAT DEFAULT NULL ;

alter table ncd20051207.SCC modify PM25fac FLOAT DEFAULT NULL ;

alter table ncd20051207.SCC modify SOAfac FLOAT DEFAULT NULL ;

alter table ncd20051207.VMTMonthAllocation modify AllocFactor FLOAT DEFAULT '0.00' ;

alter table ncd20051207.County change

CountyVMTMonthAllocationDataSource

CountyVMTMonthAllocDataSource smallint NOT NULL DEFAULT '0' ;

To assure that data is available for modeling Broomfield County CO (08014) in 2001, countyyear and countyyearmonth entries were inserted for that year to match Boulder County (08013).

# Change Log for NCD20050928

rr added to Filetype table with the following script:

# MySQL script to add rail equipment to the filetype table

use ncd20050928;

insert into filetype

(filetypeid,filetypedescription)

values ('rr','Railway maintenance equipment')

;

# Change Log for NCD20050831

This database works with NR05. It will not work with NR04, because the SCC for hydraulic power units has changed from xxxxxx5050 to xxxxxx6035 and segment changed from 5 to 6.

Three changes were made:

1.Perl script was run to change all hydraulic power unit codes in the external files.

2. Script run to change SCC's in the SCC table. Last four digits from 5050 to 6035.

3. Script run to change segment from 5 to 6 (agricultural to commercial).

**Perl script to change all hydraulic power unit codes in the external files:**

# FixExternalFiles.plx

# Script to fix external files for NR05

# Substitute corrected SCC for hydraulic power units

$oldncd='ncd20050714';

$newncd='ncd20050831';

$dbpath="D:\\mysql\\data";

$oldextfilespath="$dbpath\\$oldncd\\externalfiles";

$newextfilespath="$dbpath\\$newncd\\externalfiles";

$count=0;

foreach $file (`dir /b $oldextfilespath`){

$count++;

print "file = $file\n";

$infile="$oldextfilespath\\$file";

print "infile = $infile\n";

$outfile="$newextfilespath\\$file";

print "outfile = $outfile\n";

open(in1,"$infile");

open(out1,">$outfile");

while (<in1>){

foreach $scc6(

'226000',

'226500',

'226700',

'227000',

'226800',

){

s/${scc6}5050/${scc6}6035/;

}

print out1;

}

close(in1);

close(out1);

# if ($count==1) {exit;};

print "count = $count\n";

}

**Script to change SCC's in the SCC table:**

# fixSCCTable.plx

# Script to fix SCC in SCCTable

# Substitute corrected SCC for hydraulic power units

$ncd='ncd20050831';

$sql="

use $ncd;

update scc set SCC=2260006035 where SCC=2260005050;

update scc set SCC=2265006035 where SCC=2265005050;

update scc set SCC=2267006035 where SCC=2267005050;

update scc set SCC=2270006035 where SCC=2270005050;

update scc set SCC=2268006035 where SCC=2268005050;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

**Script to change segment in SCC table:**

# fixSCCTable.plx

# Script to fix segment in SCCTable

# Substitute corrected segment for hydraulic power units

$ncd='ncd20050831';

$sql="

use $ncd;

update scc set segment=6 where SCC=2260006035;

update scc set segment=6 where SCC=2265006035;

update scc set segment=6 where SCC=2267006035;

update scc set segment=6 where SCC=2270006035;

update scc set segment=6 where SCC=2268006035;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

# Change Log for NCD20050714

MySQL scripts were run to add and populate and the ORNR field to SCC. This field has values of OR or NR, indicating whether the SCC is for on-road vehicles or non-road equipment. These scripts are:

# addORNRToSCC.sql

# Script to add on‑road/non‑road to SCC table

use NCD20050714;

alter table scc

add column (

ORNR char(2) not null

)

;

# setORNRInSCC.sql

# Script to set on‑road/non‑road in SCC table

use NCD20050714;

update scc

set ORNR = if(segment=0,'OR','NR')

;

A MySQL script was run to correct the NH3 emission factors in the SCC table:

# NH3.sql

# Script to fix NH3 in SCC

use NCD20050714;

update scc

set mgNH3perGallon=

case fueltype

when 'Dies' then 83.3

when 'Gas' then 116.

else null

end

where ORNR='NR'

;

update scc

set mgNH3perGallon=null

where ORNR='OR'

;

A MySQL script was run to add a table called cntycnts, with fields fipsstateid and NumCounties, which is the number of counties in the state. This table is used for completeness checks of NMIM runs.

use ncd20050714;

drop table if exists cntycnts;

create table cntycnts

select fipsstateid,count(distinct fipsstateid,fipscountyid) as NumCounties

from county

group by fipsstateid

;

New temperature and humidity values were obtained from Air Improvement Resources (AIR). The original values supplied by AIR were improperly calculated, so that relative humidity values could exceed 100%. The new values also include some new temperature and humidity measurements that were not included in the original calculations, so that some averaged temperatures are also changed. Although new values were obtained for calendar years 1999 through 2003 and 30 year averages, only the 30 year averages and the 2002 calendar year temperature and humidity values were loaded into the database. A new DataSourceID (625) was created to indicate the new updated temperature and humidity values.

The data obtained from AIR does not include Broomfield County, Colorado (14). Broomfield is assumed to have the same temperature and humidity values as Boulder County (13). This data was duplicated for Broomfield and appended to the tables.

delete from ncd20050714.countyyearmonthhour

where year=2002;

insert into ncd20050714.countyyearmonthhour

(fipscountyid,fipsstateid,year,month,hourid,

temperature,relativehumidity)

select fipscountyid,fipsstateid,"2002",month,hourid,

temperature,relativehumidity

from temperature.countymonthhour2002;

# Add Broomfield County, CO.

insert into ncd20050714.countyyearmonthhour

(fipscountyid,fipsstateid,year,month,hourid,

temperature,relativehumidity)

select "14",fipsstateid,"2002",month,hourid,

temperature,relativehumidity

from temperature.countymonthhour2002

where fipsstateid=8 and fipscountyid=13;

truncate ncd20050714.countymonthhour;

insert into ncd20050714.countymonthhour

(fipscountyid,fipsstateid,month,hourid,averagetemp,

averagerelhumidity,datasourceid)

select fipscountyid,fipsstateid,month,hourid,

temperature,relativehumidity,"625"

from temperature.countymonthhournew;

# Add Broomfield County, CO.

insert into ncd20050714.countymonthhour

(fipscountyid,fipsstateid,month,hourid,averagetemp,

averagerelhumidity,datasourceid)

select "14",fipsstateid,month,hourid,

temperature,relativehumidity,"625"

from temperature.countymonthhournew

where fipsstateid=8 and fipscountyid=13;

All of the I/M program description files, both those submitted by states and those previously created for EPA were reviewed. Many changes were made in these external data files, some new I/M program files were made and all files were made consistent with the NMIM file naming conventions. The changes to the files are documented separately. The CountyYear table was updated to appropriately use the new files and new file names. The old I/M program description files were deleted from the externalfiles subdirectory of the database and the new renamed files were placed there. These are the changes to the CountyYear table:

update countyyear set imfilename="0202099.imp" where fipsstateid=2 and imfilename="ak85a.im";

update countyyear set imfilename="0209099.imp" where fipsstateid=2 and imfilename="ak85f.im";

update countyyear set imfilename="0401395.imp" where fipsstateid=4 and imfilename="az95p.im" and year<2002;

update countyyear set imfilename="0401301.imp" where fipsstateid=4 and imfilename="az95p.im" and year>=2002;

update countyyear set imfilename="0401301.imp" where fipsstateid=4 and imfilename="0401302.imp" and year>=2002;

update countyyear set imfilename="0401902.imp" where fipsstateid=4 and imfilename="az95t.im";

update countyyear set imfilename="0600199.imp" where fipsstateid=6 and imfilename="ca95b.im";

update countyyear set imfilename="0607999.imp" where fipsstateid=6 and imfilename="ca99e.im";

update countyyear set imfilename="0800199.imp" where fipsstateid=8 and imfilename="co95d.im";

update countyyear set imfilename="0804199.imp" where fipsstateid=8 and imfilename="co95c.im";

update countyyear set imfilename="0900199.imp" where fipsstateid=9 and imfilename="ct98.im";

update countyyear set imfilename="1000191.imp" where fipsstateid=10 and fipscountyid=1;

update countyyear set imfilename="1000383.imp" where fipsstateid=10 and fipscountyid=3;

update countyyear set imfilename="1000591.imp" where fipsstateid=10 and fipscountyid=5;

update countyyear set imfilename="1100102.imp" where fipsstateid=11 and imfilename="dc99.im" and year<2003;

update countyyear set imfilename="1100103.imp" where fipsstateid=11 and imfilename="dc99.im" and year>=2003;

update countyyear set imfilename="1200099.imp" where fipsstateid=12 and imfilename="fl92.im";

update countyyear set imfilename="1305799.imp"

where fipsstateid=13 and fipscountyid in (57,63,77,97,113,117,151,223,247) and year=1999;

update countyyear set imfilename="1305700.imp"

where fipsstateid=13 and fipscountyid in (57,63,77,97,113,117,151,223,247) and year in (2000,2001);

update countyyear set imfilename="1305702.imp"

where fipsstateid=13 and fipscountyid in (57,63,77,97,113,117,151,223,247) and year>=2002;

update countyyear set imfilename="1306799.imp"

where fipsstateid=13 and fipscountyid in (67,89,121,135) and year=1999;

update countyyear set imfilename="1306700.imp"

where fipsstateid=13 and fipscountyid in (67,89,121,135) and year in (2000,2001);

update countyyear set imfilename="1306702.imp"

where fipsstateid=13 and fipscountyid in (67,89,121,135) and year>=2002;

update countyyear set imfilename="1600099.imp" where fipsstateid=16 and imfilename="id85.im";

update countyyear set imfilename="1700099.imp" where fipsstateid=17 and imfilename="il99.im" and year<2002;

update countyyear set imfilename="1700002.imp" where fipsstateid=17 and imfilename="il99.im" and year>=2002;

update countyyear set imfilename="1700099.imp" where fipsstateid=17 and fipscountyid=133 and year<2002;

update countyyear set imfilename="1700002.imp" where fipsstateid=17 and fipscountyid=133 and year>=2002;

update countyyear set imfilename="1806101.imp" where fipsstateid=18 and imfilename="in01.im";

update countyyear set imfilename="1808997.imp" where fipsstateid=18 and imfilename="in97.im";

update countyyear set imfilename="2111198.imp" where fipsstateid=21 and imfilename="ky98.im" and year<2002;

update countyyear set imfilename="2111102.imp" where fipsstateid=21 and imfilename="ky98.im" and year>=2002;

update countyyear set imfilename="2200000.imp" where fipsstateid=22 and imfilename="la00.im";

update countyyear set imfilename="2300599.imp" where fipsstateid=23 and imfilename="me99.im";

update countyyear set imfilename="2400002.imp" where fipsstateid=24 and fipscountyid in (3,5,13,25,27,31,33,43,510) and year<2003;

update countyyear set imfilename="2400003.imp" where fipsstateid=24 and fipscountyid in (3,5,13,25,27,31,33,43,510) and year>=2003;

update countyyear set imfilename="2400995.imp" where fipsstateid=24 and fipscountyid in (9,15,17,21,35) and year<2003;

update countyyear set imfilename="2400903.imp" where fipsstateid=24 and fipscountyid in (9,15,17,21,35) and year>=2003;

update countyyear set imfilename="2500099.imp" where fipsstateid=25 and imfilename="ma95.im" and year=1999;

update countyyear set imfilename="2500000.imp" where fipsstateid=25 and imfilename="ma95.im" and year in (2000,2001,2002);

update countyyear set imfilename="2500000.imp" where fipsstateid=25 and imfilename="2500002.imp" and year in (2000,2001,2002);

update countyyear set imfilename="2500003.imp" where fipsstateid=25 and imfilename="ma95.im" and year>=2003;

update countyyear set imfilename="2700099.imp" where fipsstateid=27 and imfilename="mn92.im";

update countyyear set imfilename="2900099.imp" where fipsstateid=29 and imfilename="mo97.im";

update countyyear set imfilename="3200099.imp" where fipsstateid=32 and imfilename="nv95.im";

update countyyear set imfilename="3300002.imp" where fipsstateid=33 and imfilename="nh02.im";

update countyyear set imfilename="3400199.imp" where fipsstateid=34 and imfilename="nj97.im" and year=1999;

update countyyear set imfilename="3400100.imp" where fipsstateid=34 and imfilename="nj97.im" and year in (2000,2001,2002,2003,2004);

update countyyear set imfilename="3400100.imp" where fipsstateid=34 and imfilename="3400102.imp" and year in (2000,2001,2002,2003,2004);

update countyyear set imfilename="3400105.imp" where fipsstateid=34 and imfilename="nj97.im" and year>=2005;

update countyyear set imfilename="3500189.imp" where fipsstateid=35 and imfilename="nm97.im" and year<2003;

update countyyear set imfilename="3500189.imp" where fipsstateid=35 and imfilename="3500102.imp" and year<2003;

update countyyear set imfilename="3500103.imp" where fipsstateid=35 and imfilename="nm97.im" and year>=2003;

update countyyear set imfilename="3600101.imp" where fipsstateid=36 and

fipscountyid in (1,3,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,49,51,53,55,57,63,65,67,69,71,73,75,

77,79,83,89,91,93,95,97,99,101,105,107,109,111,113,115,117,121,123) and year<2003;

update countyyear set imfilename="3600103.imp" where fipsstateid=36 and imfilename="ny01.im" and year>=2003;

update countyyear set imfilename="3600599.imp" where fipsstateid=36 and imfilename="ny99.im" and year<2003;

update countyyear set imfilename="3600599.imp" where fipsstateid=36 and imfilename="3600502.imp" and year<2003;

update countyyear set imfilename="3600503.imp" where fipsstateid=36 and imfilename="ny99.im" and year>=2003;

update countyyear set imfilename="3702501.imp" where fipsstateid=37 and imfilename="nc01.im";

update countyyear set imfilename="3705792.imp" where fipsstateid=37 and imfilename="nc92.im";

update countyyear set imfilename="3711983.imp" where fipsstateid=37 and imfilename="nc83.im";

update countyyear set imfilename="3718387.imp" where fipsstateid=37 and imfilename="nc87.im";

update countyyear set imfilename="3905596.imp" where fipsstateid=39 and imfilename="oh96c.im" and year<2003;

update countyyear set imfilename="3905503.imp" where fipsstateid=39 and imfilename="oh96c.im" and year>=2003;

update countyyear set imfilename="3910398.imp" where fipsstateid=39 and imfilename="oh98cl.im" and year<2003;

update countyyear set imfilename="3910303.imp" where fipsstateid=39 and imfilename="oh98cl.im" and year>=2003;

update countyyear set imfilename="4102997.imp" where fipsstateid=41 and imfilename="or98.im" and year<2001;

update countyyear set imfilename="4102901.imp" where fipsstateid=41 and imfilename="or98.im" and year>=2001;

update countyyear set imfilename="4102901.imp" where fipsstateid=41 and imfilename="4102902.imp" and year>=2001;

update countyyear set imfilename="4100597.imp" where fipsstateid=41 and imfilename="or98p.im" and year<2001;

update countyyear set imfilename="4100501.imp" where fipsstateid=41 and imfilename="or98p.im" and year>=2001;

update countyyear set imfilename="4100501.imp" where fipsstateid=41 and imfilename="4100502.imp" and year>=2001;

update countyyear set imfilename="4100901.imp" where fipsstateid=41 and fipscountyid in (9,71) and year>=2001;

update countyyear set imfilename="4201797.imp" where fipsstateid=42 and imfilename="pa98ph.im";

update countyyear set imfilename="4201101.imp" where fipsstateid=42 and imfilename="pa01ole.im";

update countyyear set imfilename="4207785.imp" where fipsstateid=42 and imfilename="pa97.im";

update countyyear set imfilename="4200397.imp" where fipsstateid=42 and imfilename="pa98pt.im";

update countyyear set imfilename="4400002.imp" where fipsstateid=44 and imfilename="ri97.im";

update countyyear set imfilename="4703785.imp" where fipsstateid=47 and imfilename="tn85.im";

update countyyear set imfilename="4703785.imp" where fipsstateid=47 and imfilename="4703702.imp";

update countyyear set imfilename="4714995.imp" where fipsstateid=47 and imfilename="tn95.im";

update countyyear set imfilename="4714995.imp" where fipsstateid=47 and imfilename="4714902.imp";

update countyyear set imfilename="4715784.imp" where fipsstateid=47 and imfilename="tn84.im";

update countyyear set imfilename="4715784.imp" where fipsstateid=47 and imfilename="4715702.imp";

update countyyear set imfilename="4803902.imp" where fipsstateid=48 and

fipscountyid in (39,71,157,167,291,339,473) and year>=2000;

update countyyear set imfilename="4814102.imp" where fipsstateid=48 and fipscountyid in (141);

update countyyear set imfilename="4820197.imp" where fipsstateid=48 and

fipscountyid in (201) and year<2002;

update countyyear set imfilename="4820102.imp" where fipsstateid=48 and

fipscountyid in (201) and year>=2002;

update countyyear set imfilename="4808500.imp" where fipsstateid=48 and

fipscountyid in (85,121) and year in (2000,2001);

update countyyear set imfilename="4808502.imp" where fipsstateid=48 and

fipscountyid in (85,121) and year>=2002;

update countyyear set imfilename="4811390.imp" where fipsstateid=48 and

fipscountyid in (113,439) and year<2002;

update countyyear set imfilename="4811302.imp" where fipsstateid=48 and

fipscountyid in (113,439) and year>=2002;

update countyyear set imfilename="4901197.imp" where fipsstateid=49 and fipscountyid=11;

update countyyear set imfilename="4903502.imp" where fipsstateid=49 and fipscountyid=35 and year<2003;

update countyyear set imfilename="4903503.imp" where fipsstateid=49 and fipscountyid=35 and year>=2003;

update countyyear set imfilename="4904986.imp" where fipsstateid=49 and fipscountyid=49 and year<2002;

update countyyear set imfilename="4904902.imp" where fipsstateid=49 and fipscountyid=49 and year>=2002;

update countyyear set imfilename="4905792.imp" where fipsstateid=49 and fipscountyid=57 and year<2002;

update countyyear set imfilename="4905702.imp" where fipsstateid=49 and fipscountyid=57 and year>=2002;

update countyyear set imfilename="5101302.imp" where fipsstateid=51 and imfilename="va98.im" and

fipscountyid in (13,59,153,510,600,610,683,685);

update countyyear set imfilename="5110702.imp" where fipsstateid=51 and imfilename="va98.im" and

fipscountyid in (107,179);

update countyyear set imfilename="5000097.imp" where fipsstateid=50 and imfilename="vt97.im" and year<2002;

update countyyear set imfilename="5000002.imp" where fipsstateid=50 and imfilename="vt97.im" and year>=2002;

update countyyear set imfilename="5301198.imp" where fipsstateid=53 and fipscountyid=11 and year<2002;

update countyyear set imfilename="5301102.imp" where fipsstateid=53 and fipscountyid=11 and year>=2002;

update countyyear set imfilename="5303382.imp" where fipsstateid=53 and fipscountyid=33 and year<2002;

update countyyear set imfilename="5303302.imp" where fipsstateid=53 and fipscountyid=33 and year>=2002;

update countyyear set imfilename="5305393.imp" where fipsstateid=53 and fipscountyid=53 and year<2002;

update countyyear set imfilename="5305302.imp" where fipsstateid=53 and fipscountyid=53 and year>=2002;

update countyyear set imfilename="5306193.imp" where fipsstateid=53 and fipscountyid=61 and year<2002;

update countyyear set imfilename="5306102.imp" where fipsstateid=53 and fipscountyid=61 and year>=2002;

update countyyear set imfilename="5306385.imp" where fipsstateid=53 and fipscountyid=63 and year<2002;

update countyyear set imfilename="5306302.imp" where fipsstateid=53 and fipscountyid=63 and year>=2002;

update countyyear set imfilename="5511794.imp" where fipsstateid=55 and fipscountyid=117 and year<2001;

update countyyear set imfilename="5511702.imp" where fipsstateid=55 and fipscountyid=117 and year>=2001;

update countyyear set imfilename="5505984.imp" where fipsstateid=55 and fipscountyid in (59,79,89,101,131,133) and year<2001;

update countyyear set imfilename="5505902.imp" where fipsstateid=55 and fipscountyid in (59,79,89,101,131,133) and year>=2001;

The state supplied changes supplied by Pechan included the addition of a single new gasoline formulation (3003) which was applied to all months for all counties in Iowa (19) for the 2002 calendar year. Upon review by EPA, it was decided to disallow this change. At least a separate winter and summer formulation must be used, especially in colder areas where RVP must be higher in winter months for proper engine starting. The CountyYearMonth table was changed to set the values for Iowa (19) for the 2002 calendar year back to the default values.

delete from ncd20050714.countyyearmonth where year=2002 and fipsstateid=19;

insert into ncd20050714.countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,HwyGasolineId,

NRGasolineId,NRDieselId,RMDieselId,HwyFuelDataSource,NRFuelDataSource)

select FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,HwyGasolineId,

NRGasolineId,NRDieselId,RMDieselId,HwyFuelDataSource,NRFuelDataSource

from county20041015.countyyearmonth

where year=2002 and

fipsstateid=19;

# Change Log for NCD20050705

State supplied changes submitted by Pechan.

This was downloaded from Pechan's ftp site. Pechan delivered with name County20050705. Renamed to NCD20050705. No change log information was added by Pechan.

WS\_FTP.LOG:

2005.07.05 15:58 B D:\ad\NEI02\NMIM Database from Pechan July 2005\County20050705.zip <‑‑ Pechan /pub/NMIM\_NCD County20050705.zip

We sent Pechan NCD20050427, so that is the starting point for this database.

**Change Log for NCD20050427**

This database is intended to replace NCD20050318

The major problem with NCD20050318 was that the HwyGasolineId and NRGasolineId fields were set to tinyint and therefore were truncated.

1. Copied NCD20050318

2. Copied county20050215.countyyearmonth into NCD20050427, restoring the CountyYearMonth data

# Change Log for NCD20050318

Note new naming convention. NCD is an acronym for "NMIM County Database."

CountyMap changed so that every county represents itself, using the following script:

use county20050223hm;

update countymap

set

HwyRepFIPSCntyID=FIPSCountyId,

NRRepFIPSCntyID=FIPSCountyId

;

Delete tables: CountyYearMonthNR, CountyYearMonthRecM, CountyMonthHourAvg

# Change Log for 02/23/2005 NMIM CountyDB database

Production Database: H:\AMD\MOBILE\NMIM\CountyDB\CountyDB20050223.zip

This version, based on the previous 5/8/2004 CountyDB20040508.zip and the ERwin design CountyDB20050223.ER1, includes the following changes.

**A. Data Change**

All data was imported into this CountyDB20050223 database from CountyDB20050215.zip.

**B. Structure Change**

The structure/schema of CountyDB20040508.zip was used with the following modification.

**1. Add Fields**

(These were missing in 20040508)

Table Field Type Null Default

countynrfile DataSourceId smallint(6) YES NULL

countyyear HasNRStateInputs char(1)

countyyear HasORStateInputs char(1)

countyyearmonth RMDieselId smallint(6) YES NULL

datasource comments varchar(255) YES NULL

**2. Different Field Definitions (Use 20041015 definitions)**

Table Field Type Null Default

countyyearmonth NGId tinyint(4) YES NULL

countyyearmonth HwyGasolineId smallint(6) YES NULL

countyyearmonth NRGasolineId smallint(6) YES NULL

datasource Author varchar(25) YES NULL

datasource Sponsor varchar(30) YES NULL

datasource DocumentId varchar(30) YES NULL

**3. Change field name DataSource to DataSourceId for 4 tables.**

ALTER TABLE BaseYearVMT CHANGE DataSource DataSourceId SMALLINT(6) NULL DEFAULT NULL \p;

ALTER TABLE VMTMonthAllocation CHANGE DataSource DataSourceId SMALLINT(6) NULL DEFAULT NULL \p;

ALTER TABLE CountyYear CHANGE DataSource DataSourceId SMALLINT(6) NULL DEFAULT NULL \p;

ALTER TABLE VMTGrowth CHANGE DataSource DataSourceId SMALLINT(6) NULL DEFAULT NULL \p;

**4. Add rules for checking tables**

Table Field Type Null Field must be one of these

county Altitude char(1) "L,H"

countyyear HasNRStateInputs char(1) "N,Y"

countyyear HasORStateInputs char(1) "N,Y"

scc FuelType char(4) NO "CNG,Gas,Dies,LPG"

scctoxics ratioType char(5) YES "VOC,Gal,PM10,Mile,PMVOC"

county OzoneSeasonStartMonth between 1 and 12 (old rule requires 6)

county OzoneSeasonEndMonth between 1 and 12 (old rule requires 8)

**5. Set these fields to be "not null"**

Table Field

filetype FileTypeDescription

countyyear ATPFileName

countyyear IMFileName

state NLEVFileName

scc SCCDesc

state StateAbbr

state StateName

scc FuelType

county CountyName

hour HourName

hpmsroadtype RoadDesc

scc Part5VClassDesc

scc RoadwayTypeDesc

scc SegmentDesc

m6vclass VClassAbbr

county NGVFractionFileName

countynrfile CountyNRFileName

countyyear AvgSpeedDistBaseFileName

countyyear DieselFractFileName

countyyear DiurnSoakActivityFileName

countyyear HotSoakFileName

countyyear MileAccumFileName

countyyear NRACTFileName

countyyear RegDistFileName

countyyear SoakDistFileName

countyyear StartDistFileName

countyyear TripLengthFileName

countyyear TripsPerDayFileName

countyyear VMTByHourFileName

state T2CertFileName

state T2EvapPhaseInFileName

state T2ExhPhaseInFileName

m6vclass VClassDesc

m6vtype VTypeDesc

# Change Log for 20050215

I started by copying County20041201 and then made the following changes:

1. Copied emissiontype table from County20040508, which contains the fifth emissiontype, which appears to have been left behind when Pechan added State inputs for the Draft 2002 NEI.

2. Changed null strings to "" for all cases for which the design was changed to disallow nulls, using the following Perl/MySQL script:

# Script to convert null strings to "".

$ncd=county20050215;

foreach $tablefield(

'filetype FileTypeDescription',

'countyyear ATPFileName',

'countyyear IMFileName',

'state NLEVFileName',

'scc SCCDesc',

'state StateAbbr',

'state StateName',

'scc FuelType',

'county CountyName',

'hour HourName',

'hpmsroadtype RoadDesc',

'scc Part5VClassDesc',

'scc RoadwayTypeDesc',

'scc SegmentDesc',

'm6vclass VClassAbbr',

'county NGVFractionFileName',

'countynrfile CountyNRFileName',

'countyyear AvgSpeedDistBaseFileName',

'countyyear DieselFractFileName',

'countyyear DiurnSoakActivityFileName',

'countyyear HotSoakFileName',

'countyyear MileAccumFileName',

'countyyear NRACTFileName',

'countyyear RegDistFileName',

'countyyear SoakDistFileName',

'countyyear StartDistFileName',

'countyyear TripLengthFileName',

'countyyear TripsPerDayFileName',

'countyyear VMTByHourFileName',

'state T2CertFileName',

'state T2EvapPhaseInFileName',

'state T2ExhPhaseInFileName',

'm6vclass VClassDesc',

'm6vtype VTypeDesc',

){

print "tablefield = $tablefield\n";

($table,$field)=split(/\t/,$tablefield);

print "table = $table\n";

print "field = $field\n";

$sql="

use $ncd;

update $table

set $field=ifnull($field,\"\")

;

";

open(out1,">script.sql");

print out1 $sql;

close(out1);

`mysql < script.sql`;

}

# Change Log for 20041201

The remaining changes from the state supplied data included in the 20041001 version provided by Pechan. The only changes from the 20041015 version occur in the CountyYearMonth table. The additional diesel sulfur values have already been added to the Diesel table in the 20041006 version and no new gasolines were added to the Gasoline table.

Changes were made to some of the fuel IDs in some of the counties in the following states:

o Colorado (FIPSStateID=8)

o Delaware (10)

o Maryland (24)

o Michigan (26)

o Virginia (51)

o Washington (53)

All other tables were copied directly from the County20041015 version.

create database if not exists county20041201;

Use county20041201;

# Rebuild CountyYearMonth, using state supplied fuel mapping from Pechan database

# and recreational marine diesel fuel mapping from the default database table:

drop table if exists CountyYearMonth;

create table CountyYearMonth

(

FIPSCountyId smallint not null,

FIPSStateId tinyint not null,

Year smallint not null,

Month tinyint not null,

NGId tinyint,

HwyDieselId smallint,

HwyGasolineId smallint,

NRGasolineId smallint,

NRDieselId smallint,

RMDieselId smallint,

HwyFuelDataSource smallint,

NRFuelDataSource smallint

)

;

insert into CountyYearMonth

(

FIPSCountyId,

FIPSStateId,

Year,

Month,

NGId,

HwyDieselId,

HwyGasolineId,

NRGasolineId,

NRDieselId,

RMDieselId,

HwyFuelDataSource,

NRFuelDataSource

)

select

n.FIPSCountyId,

n.FIPSStateId,

n.Year,

n.Month,

n.NGId,

n.HwyDieselId,

n.HwyGasolineId,

n.NRGasolineId,

n.NRDieselId,

r.NRDieselId,

n.HwyFuelDataSource,

n.NRFuelDataSource

from

county20041001.CountyYearMonth as n

inner join

county20040508e.CountyYearMonthRecm as r

using (FIPSCountyId, FIPSStateId, Year, Month)

;

alter table CountyYearMonth

add primary key

(

FIPSCountyId,

FIPSStateId,

Year,

Month

)

;

# Fix problems identified in the state supplied data.

# Use default nonroad diesel fuels for Colorado.

delete from countyyearmonth

where fipsstateid=8 and

year=2002;

insert into CountyYearMonth

(

FIPSCountyId,

FIPSStateId,

Year,

Month,

NGId,

HwyDieselId,

HwyGasolineId,

NRGasolineId,

NRDieselId,

RMDieselId,

HwyFuelDataSource,

NRFuelDataSource

)

select

n.FIPSCountyId,

n.FIPSStateId,

n.Year,

n.Month,

n.NGId,

n.HwyDieselId,

n.HwyGasolineId,

n.NRGasolineId,

r.NRDieselId,

s.NRDieselId,

n.HwyFuelDataSource,

n.NRFuelDataSource

from

county20041001.CountyYearMonth as n,

county20040508e.CountyYearMonthNR as r,

county20040508e.CountyYearMonthRecM as s

where n.fipsstateid=r.fipsstateid and

n.fipsstateid=s.fipsstateid and

n.fipscountyid=r.fipscountyid and

n.fipscountyid=s.fipscountyid and

n.year=r.year and n.year=s.year and

n.month=r.month and n.month=s.month and

n.fipsstateid=8 and

n.year=2002

;

# Set the nonroad gasolines in Washington to be the same as highway gasoline.

update countyyearmonth

set nrgasolineid=hwygasolineid

where fipsstateid=53 and

fipscountyid in (35,53,63) and

year=2002;

# Broomfield 2002 fuel data assumed to be the same as for Boulder County (13).

delete from countyyearmonth where fipsstateid=8 and fipscountyid=14;

insert into countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,HwyGasolineId,

NRGasolineId,NRDieselId,RMDieselId,HwyFuelDataSource,NRFuelDataSource)

select 14,FIPSStateId,Year,Month,NGId,HwyDieselId,HwyGasolineId,

NRGasolineId,NRDieselId,RMDieselId,HwyFuelDataSource,NRFuelDataSource

from county20041015.countyyearmonth

where fipsstateid=8 and fipscountyid=13;

# Change Log for 20041015

Three of the state supplied NONROAD files were scrambled, and would not run. New (unscrambled) versions of the files were obtained from Pechan:

o 18000.pop

o 39000.pop

o 55000.pop

The WOB file for Washington state (53000wob.alo) did not run.

Broomfield County, Colorado (08014), was created in 2001 from parts of Adams (1, 4.188%), Boulder (13, 7.372%), Jefferson (59, 0.294%) and Weld (123, 0.0055%) Counties and needs to be added to all of the NMIM tables. This will allow NMIM to model Broomfield. However, NONROAD cannot handle counties which are not internally coded and will require updating all of the population and allocation files for Colorado.

use county20041015;

# Add Broomfield County, Colorado, to all of the NMIM tables.

# County20041001 had values for Broomfield County. Some values were passed into

# County20041006, including the CountyNRFiles, CountyYear and BaseYearVMT tables.

# Values in County, CountyYearMonth and CountyMonthHour tables were not passed.

#

# Although there is a state supplied NONROAD file (08000oil.alo), it does \*not\* contain

# allocations for Broomfield County (08014).

delete from county where fipsstateid=8 and fipscountyid=14;

insert into county

(FIPSCountyId,FIPSStateId,CountyName,Altitude,BarometricPressure,

OzoneSeasonStartMonth,OzoneSeasonStartDay,OzoneSeasonEndMonth,

OzoneSeasonEndDay,Stage2StartYear,PhaseInYears,LDVStage2Percent,

HDVStage2Percent,NGVFractionFileName)

select FIPSCountyId,FIPSStateId,CountyName,Altitude,BarometricPressure,

OzoneSeasonStartMonth,OzoneSeasonStartDay,OzoneSeasonEndMonth,

OzoneSeasonEndDay,Stage2StartYear,PhaseInYears,LDVStage2Percent,

HDVStage2Percent,NGVFractionFileName

from county20041001.county

where fipsstateid=8 and fipscountyid=14;

# Broomfield 2002 fuel data assumed to be the same as for Boulder County (13).

delete from countyyearmonth where fipsstateid=8 and fipscountyid=14;

insert into countyyearmonth

(FIPSCountyId,FIPSStateId,Year,Month,NGId,HwyDieselId,HwyGasolineId,

NRGasolineId,NRDieselId,RMDieselId,HwyFuelDataSource,NRFuelDataSource)

select 14,FIPSStateId,Year,Month,NGId,HwyDieselId,HwyGasolineId,

NRGasolineId,NRDieselId,RMDieselId,HwyFuelDataSource,NRFuelDataSource

from county20041006.countyyearmonth

where fipsstateid=8 and fipscountyid=13;

delete from countymonthhour where fipsstateid=8 and fipscountyid=14;

insert into countymonthhour

(FIPSCountyId,FIPSStateId,Month,HourID,AverageTemp,AverageRelHumidity,DataSourceId)

select FIPSCountyId,FIPSStateId,Month,HourID,AverageTemp,AverageRelHumidity,DataSourceId

from county20041001.countymonthhour

where fipsstateid=8 and fipscountyid=14;

# 2002 hourly temperatures for Broomfield County are the same as for 30 year average.

delete from countyyearmonthhour where fipsstateid=8 and fipscountyid=14;

insert into countyyearmonthhour

(FIPSCountyId,FIPSStateId,Year,Month,HourID,Temperature,RelativeHumidity)

select FIPSCountyId,FIPSStateId,2002,Month,HourID,averagetemp,averageRelHumidity

from county20041001.countymonthhour

where fipsstateid=8 and fipscountyid=14;

delete from countymonth where fipsstateid=8 and fipscountyid=14;

insert into countymonth

(fipscountyid,fipsstateid,month)

values (14,8,1),

(14,8,2),

(14,8,3),

(14,8,4),

(14,8,5),

(14,8,6),

(14,8,7),

(14,8,8),

(14,8,9),

(14,8,10),

(14,8,11),

(14,8,12) ;

# Set the Broomfield representing county to be Boulder County.

# Broomfield County was created in 2001 from parts of Adams (1),

# Boulder (13), Jefferson (59) and Weld (123) Counties.

delete from countymap where fipsstateid=8 and fipscountyid=14;

insert into countymap

(fipscountyid,fipsstateid,hwyrepfipscntyid,nrrepfipscntyid)

values (14,8,13,13);

# Change Log for 20041006

This version began with the 20041001 version of the database from Pechan. All of the changes listed for the 20040508e version (described below) were made to this version as well. The script used to add the rmdieselid field to the CountyYearMonth table used only values from the 20040508 version of the database. As a result, none of the state supplied changes to the CountyYearMonth table (highway and nonroad gasoline and diesel fuel properties) were used. The two additional fields (for Task 35) had already been added by Pechan and were already populated.

Some further changes needed to be made to the external data files in order to be run using NMIM.

o All of the default external data files were added to the directory containing the new state-provided files submitted by Pechan.

o File 41005202.imp was renamed to 41005202.imc.

o The I/M data file 1700002.imp was altered from using the default I/M cutpoint file name (Mycuts.d) to using the file 1700202.imc, containing the same information as the default Mycuts.d file, which was added to the external data files directory.

o All of the Ohio external data files for registration distributions (39\*.reg) were altered to add the necessary header (REG DIST).

o All of the Texas and Virginia external data files for diesel sales fractions (48\*.dsf and 51\*.dsf) and file 3600502.dsf were altered to add the appropriate header (2002) and remove any text. No text is allowed in this file.

o The Jackson County, Kentucky, I/M program file (2111102.imp) contained anti-tampering program information. This is not allowed in an I/M program data file. The rows with ATP data were changed to comments.

o All of the default NLEVFileName values in the State table were renamed using the file naming convention and the external data files were appropriately renamed.

The NLEVFileName values changed in the State table were made with the following SQL:

# Rename default NLEV data files to naming convention.

Update state set nlevfilename="6000.nlv" where fipsstateid=6;

Update state set nlevfilename="42000.nlv" where fipsstateid in (9,33,34,42);

Update state set nlevfilename="23000.nlv" where fipsstateid=23;

Update state set nlevfilename="50000.nlv" where fipsstateid=50;

All of the values in the CountyMonthHour and CountyYearMonthHour tables were replaced with new values provided by AIR which utilize more of the NCDC temperature sites and calculate the real average daily minimum and maximum values. None of the state supplied temperature data was used.

The Stage 2 fields of the County table (Stage2StartYear, PhaseInYears, LDVStage2Percent and HDVStage2Percent), which were not populated by state supplied data from Pechan, were populated from the survey of EPA Regional Offices, based on the 1999 NEI list of Stage 2 counties. The seven counties provided by Washington state with Stage 2 program start years had null values for PhaseInYears, LDVStage2Percent and HDVStage2Percent. These were populated with default values of 1 year phase-in and 86% effectiveness.

The SCCToxics table had several small errors in the factors used to generate toxic emission rates from HC results. These are fixed by the following script:

# 2,2,4‑Trimethylpentane

update scctoxics set evapbasegas=0.0157667,

evapethgas =0.0157667,

evapmtbegas=0.0157667,

evaprfggas =0.0157667 where pollutantcodeid=30;

# Benzene

update scctoxics set evapbasegas=0.0220,

evapethgas =0.01254,

evapmtbegas=0.01584,

evaprfggas =0.01584 where pollutantcodeid=41;

# Ethyl Benzene

update scctoxics set evapbasegas=0.0077,

evapethgas =0.0045,

evapmtbegas=0.0063,

evaprfggas =0.0063 where pollutantcodeid=1;

# Hexane

update scctoxics set evapbasegas=0.0234,

evapethgas =0.0096,

evapmtbegas=0.0087,

evaprfggas =0.0087 where pollutantcodeid=6;

# Naphthalene

update scctoxics set evapbasegas=0.0004,

evapethgas =0.0004,

evapmtbegas=0.0004,

evaprfggas =0.0004 where pollutantcodeid=52;

# Toluene

update scctoxics set evapbasegas=0.0413,

evapethgas =0.0195,

evapmtbegas=0.0276,

evaprfggas =0.0276 where pollutantcodeid=5;

# Xylene

update scctoxics set evapbasegas=0.0223,

evapethgas =0.0119,

evapmtbegas=0.0188,

evaprfggas =0.0188 where pollutantcodeid=10;

All of the tables in the database which are not needed for execution or documentation of the active tables were removed from the database. These tables are retained in the County20041001 version of the database as documentation of the changes from the County20040508 version.

# Eliminate any unneccessary tables from the final database.

drop table if exists countyyearmonthnr;

drop table if exists countyyearmonthrecm;

drop table if exists baseyearvmt\_audit;

drop table if exists county\_audit;

drop table if exists countymonthhour\_audit;

drop table if exists countynrfile\_audit;

drop table if exists countyvmtmonthallocation\_audit;

drop table if exists countyyear\_audit;

drop table if exists countyyear\_deleted;

drop table if exists countyyearmonth\_audit;

drop table if exists datasource\_audit;

drop table if exists diesel\_audit;

drop table if exists state\_audit;

The NMIM20040920 version cannot handle NULL values in the ATPFileName or IMFileName fields of the CountyYear table. All NULL values must be changed to blanks.

# Eliminate NULL values from atpfilename and imfilename fields in the CountyYear table.

update countyyear set atpfilename="" where isnull(atpfilename)=1;

update countyyear set imfilename="" where isnull(imfilename)=1;

# Change Log for 20041001

This version was provided by Pechan and Associates. This version began with the 20040508 version. For Task 35, two fields were added to CountyYear:

# Script to add two fields to CountyYear table of NCD.

use county20040508e;

alter table CountyYear

add column (

HasNRStateInputs char(1) not null default 'N',

HasORStateInputs char(1) not null default 'N'

)

Pechan populated these two fields, depending on whether state supplied inputs affecting the NONROAD modeling or onroad (MOBILE) modeling were provided.

The state supplied data was added to the State, County, CountyYear, CountyNRFile, BaseYearVMT, CountyVMTMonthAllocation, Diesel and CountyYearMonth tables. The external data files indicated in the tables were also added to the external data files directory of the database.

All of the changes to tables are documented in a set of additional tables added to the database with the "Audit" extension. These tables indicate which records in each table which were altered or deleted.

# Change Log for 20040508e

This version is the same as 20040508, except for the following:

For Task 37, VMTGrowth has been truncated. MySQL script: truncate table vmtgrowth;

For Task 35, two fields added to CountyYear:

# Script to add two fields to CountyYear table of NCD.

use county20040508e;

alter table CountyYear

add column (

HasNRStateInputs char(1) not null default 'N',

HasORStateInputs char(1) not null default 'N'

)

;

For Task 38, Rebuild CountyYearMonth, according to the following script:

use county20040508e

drop table if exists CountyYearMonth;

create table CountyYearMonth

(

FIPSCountyId smallint not null,

FIPSStateId tinyint not null,

Year smallint not null,

Month tinyint not null,

NGId tinyint,

HwyDieselId smallint,

HwyGasolineId smallint,

NRGasolineId smallint,

NRDieselId smallint,

RMDieselId smallint,

HwyFuelDataSource smallint,

NRFuelDataSource smallint

)

;

insert into CountyYearMonth

(

FIPSCountyId,

FIPSStateId,

Year,

Month,

NGId,

HwyDieselId,

HwyGasolineId,

NRGasolineId,

NRDieselId,

RMDieselId,

HwyFuelDataSource,

NRFuelDataSource

)

select

n.FIPSCountyId,

n.FIPSStateId,

n.Year,

n.Month,

n.NGId,

n.HwyDieselId,

n.HwyGasolineId,

n.NRGasolineId,

n.NRDieselId,

r.NRDieselId,

n.HwyFuelDataSource,

n.NRFuelDataSource

from

CountyYearMonthNR as n

inner join CountyYearMonthRecm as r

using (FIPSCountyId, FIPSStateId, Year, Month)

;

alter table CountyYearMonth

add primary key

(

FIPSCountyId,

FIPSStateId,

Year,

Month

)

;

# Change Log for 05/08/2004 NMIM CountyDB database

Production Database: H:\AMD\MOBILE\NMIM\CountyDB\County20040508.zip

This version, based on the previous 2/18/2004 version and the ERwin design CountyDB20040508.ER1, includes the following changes.

*Changes in Table structure:*

1. The VMTGrowthRate field in the VMTGrowth table must be changed to a float.

2. The VMT field in the BaseYearVMT table must be changed to a float.

3. The HwyDieselId and NRDieselId fields in the CountyYearMonth table must be changed from tinyint(2) to smallint.

4. The DieselId field in the Diesel table must be changed to smallint.

*Changes to Values in Tables:*

PollutantCode

The PollutantCode field value of "NOx" in the PollutantCode table must be changed to "NOX" to be consistent with NIF3.

UPDATE PollutantCode SET PollutantCode=”NOX” WHERE PollutantCode=”NOx” \p;

CountyMonthHour

Replace all old values with new values.

The values supplied by the contractor (AIR) were rounded to integers in the spreadsheet they provided. The new values have one more significant digit (one decimal).

CountyYearMonthHour

Add the 1999, 2000, 2001 and 2003 calendar year data from this table.

The database already has the 2002 data. All of the data was obtained from AIR.

Diesel

Replace all old values with new values.

Many new fuels were added and all of the DieselId values were changed. The changes are discussed in the CountyYearMonth table changes. All DieselId values were changed to be the same as the integer diesel sulfur content value to make it easier to add new diesel sulfur values. Since there are now more than 99 sulfur values, the DieselSulfur field was changed from a tinyint to a smallint.

CountyYearMonth

Replace all old values with new values.

A set of nonroad diesel sulfur values used in recent nonroad rule analysis was provided by Craig Harvey. These values were used to replace all previous values in all calendar years. A separate set of diesel sulfur values specifically for nonroad recreational marine was used to create a separate table.

The 2000 Alliance fuel survey data was used by Pechan to develop county specific diesel sulfur values for highway vehicles. These values are used for all calendar years from 1999 through May of 2006. Starting in June of 2006, new highway diesel sulfur values are phased in based on estimates of the effects of new rules provided by Dave Korotney. California moves to 11 ppm in June 2006. Non-California moves to 43 ppm in June 2006 and then to 11 ppm in June 2010. Since there are now more than 99 sulfur values, the HwyDieselId and NRDieselID fields were changed from a tinyint to a smallint.

The gasolines used by California, New York, Connecticut and Arizona between 2004 and 2006 were changed to reflect the phase out of MTBE. These fuel descriptions were obtained through Rich Cook, from Chris Brunner and Lester Wyborny.

This file is identical to the CountyYearMonthNR table added below.

VMTGrowth

Replace all old values with new values.

Warning : This file has been reduced to only include growth rates for the 1999 through 2002 calendar years. The growth rates were calculated using the new 1999 base year VMT and the 2002 VMT from Pechan. All growth rates for categories with zero VMT in 1999 are zero, even if there is non-zero VMT in calendar year 2002. All 1999 growth rates are zero.

BaseYearVMT

Replace all old values with new values.

Only base years 1999 and 2002 are included. Base year 1999 contains all new values recently obtained from Pechan.

Gasoline

Add the new gasoline descriptions from this table.

Many new gasoline descriptions have been added to account for the MTBE phase out in California, New York, Connecticut and Arizona.

DataSource

Replace all old values with new values.

Many new data sources were added and obsolete data sources were deleted to reflect the changes in other tables.

CountyMap

Replace all old values with new values.

This map applies only to 2002 calendar year and was provided by Harvey Michaels.

*Tables to be added to the database:*

CountyYearMonthRecM

Same as CountyYearMonth, but with the nonroad recreational marine diesel sulfur values.

CountyYearMonthNR

Same as CountyYearMonth, but with the diesel sulfur values for all equipment except recreational marine.

CountyMonthHourAvg

Same as CountyMonthHour, used as a backup for special modeling.

# Change Log for 02/18/2004 NMIM CountyDB database

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20040218.zip

This version, based on the previous 1/29/2004 version, includes the following changes. It is intended to be used for Cimulus Task 30 and later.

**Change 1)** Loaded the new temperature data provided by David Brzezinski into the CountyYearMonthHour table.

**Change 2)** Replaced the barometric pressure values in the County table with the values of average barometric pressure provided by David Brzezinski (the average pressure is the straight average of all hours in all months for each county).

**Change 3)** Eliminated cases in the county database where the market share is greater than zero, but the oxygen content of the gasoline is zero. This causes MOBILE6 errors.

update gasoline set etohmktshare=0 where etohmktshare>0 and etohvolume=0;

update gasoline set mtbemktshare=0 where mtbemktshare>0 and mtbevolume=0;

update gasoline set etbemktshare=0 where etbemktshare>0 and etbevolume=0;

update gasoline set tamemktshare=0 where tamemktshare>0 and tamevolume=0;

**Change 4)** Dropped the fields TotEtherMktShare and TotEtherVolume in the Gasoline table. They are redundant and never used by MOBILE6 or NONROAD.

alter table gasoline drop totethermktshare;

alter table gasoline drop totethervolume;

**Change 5)** Deleted temporary tables, County1999, County2002, CountyBase, CountyMonth1999, CountyMonth2002, and CountyMonthAvg.

# Change Log for 01/29/2004 NMIM CountyDB database

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20040129.zip

This version, based on the previous 1/21/2004 version, includes the following changes. It is intended to be used for Cimulus Task 30 and later.

**Change 1)** Added a new description field “HourName” in Hour table.

**Change 2)** Kept empty table CountyYearMonthHour. (data of years 1999 and 2000 in 1/23/2004 version was not loaded in this version per David B’s suggestion)

**Change 3)** Added barometric pressure information in County table from Dave B. (data was copied from the County table of 01/23/2004 version)

# Change Log for 01/23/2004 NMIM CountyDB database

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20040123.zip

This version was the one actually sent to Cimulus for Task 30.

Further changes were:

Removal of alternative versions of CountyMap and County that would be confusing to Cimulus.

Addition of barometric pressure information in County table by Dave B.

Population of CountyYearMonthHour table for 1999 and 2000 by Dave B.

# Change Log for 01/21/2004 NMIM CountyDB database (a.k.a. Expanded County database)

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20040121.EXE

This version, an expanded County database based on the previous 12/04/2003 version, has major structure modifications (see rtf files in readme folder for details) and includes the following changes. It is intended to be used for Cimulus Task 30 and later.

**Change 1)** Added new tables: CountyNRFile, FileType, CountyMap, CountyVMTMonthAllocation, CountyMonthHour, Hour, and CountyYearMonthHour.

**Change 2)** Loaded the average temperature and humidity data into table CountyMonthHour.

**Change 3)** Made data changes to VMTGrowth. (suggested by Dave B.)

**Change 4)** Removed Hg and As from the NMIMOutput PollutantCode and SCCToxics tables.

**Change 5)** Dropped table Gasoline2 , which also means the deletion of the two foreign key fields in HwyGasolineA and HwyGasolineB in CountyYearMonth.

**Change 6)** Dropped AverageSpeed table.

# Change Log for 12/04/2003 CountyDB database

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20031204.EXE

This version, based on the previous 11/25/2003 version, includes the following changes and is intended to be used for Cimulus Task 29 and later.

**Change 1)** Added a new table County2002.

**Change 2)** Updated County and County1999 tables because the representing counties have changed due to the changes to fuels and temperatures.

Updates to County table:

FIPSStateId FIPSCountyId Old county.NRRepFIPSCntyId New county.NRRepFIPSCntyId

------------ --------------- ------------------------------- ---------------------------------

6 15 3 15

6 23 3 15

6 45 3 15

6 53 3 15

6 79 3 15

6 83 3 15

6 87 3 15

6 41 1 41

6 75 1 41

6 81 1 41

6 97 1 41

Updates to County1999 table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIPSStateId | FIPSCountyId | Old county1999  NRRepFIPSCntyId | New county1999  NRRepFIPSCntyId | |
| 6 | 15 | 3 | 15 |
| 6 | 23 | 3 | 15 |
| 6 | 45 | 3 | 15 |
| 6 | 53 | 3 | 15 |
| 6 | 79 | 3 | 15 |
| 6 | 83 | 3 | 15 |
| 6 | 87 | 3 | 15 |
| 6 | 41 | 1 | 41 |
| 6 | 75 | 1 | 41 |
| 6 | 81 | 1 | 41 |
| 6 | 97 | 1 | 41 |
| 41 | 13 | 3 | 1 |
| 41 | 17 | 3 | 1 |
| 41 | 21 | 3 | 1 |
| 41 | 23 | 3 | 1 |
| 41 | 25 | 3 | 1 |
| 41 | 27 | 3 | 1 |
| 41 | 31 | 3 | 1 |
| 41 | 37 | 3 | 1 |
| 41 | 45 | 3 | 1 |
| 41 | 49 | 3 | 1 |
| 41 | 55 | 3 | 1 |
| 41 | 59 | 3 | 1 |
| 41 | 61 | 3 | 1 |
| 41 | 63 | 3 | 1 |
| 41 | 65 | 3 | 1 |
| 41 | 69 | 3 | 1 |
| 41 | 35 | 5 | 35 |
| 53 | 3 | 3 | 1 |
| 53 | 5 | 3 | 1 |
| 53 | 7 | 3 | 1 |
| 53 | 13 | 3 | 1 |
| 53 | 17 | 3 | 1 |
| 53 | 19 | 3 | 1 |
| 53 | 21 | 3 | 1 |
| 53 | 23 | 3 | 1 |
| 53 | 25 | 3 | 1 |
| 53 | 37 | 3 | 1 |
| 53 | 39 | 3 | 1 |
| 53 | 43 | 3 | 1 |
| 53 | 47 | 3 | 1 |
| 53 | 51 | 3 | 1 |
| 53 | 65 | 3 | 1 |
| 53 | 71 | 3 | 1 |
| 53 | 75 | 3 | 1 |
| 53 | 77 | 3 | 1 |
| 53 | 9 | 3 | 9 |
| 53 | 15 | 3 | 9 |
| 53 | 27 | 3 | 9 |
| 53 | 31 | 3 | 9 |
| 53 | 35 | 3 | 9 |
| 53 | 41 | 3 | 9 |
| 53 | 45 | 3 | 9 |
| 53 | 49 | 3 | 9 |
| 53 | 53 | 3 | 9 |
| 53 | 55 | 3 | 9 |
| 53 | 57 | 3 | 9 |
| 53 | 59 | 3 | 9 |
| 53 | 67 | 3 | 9 |
| 53 | 69 | 3 | 9 |
| 53 | 73 | 3 | 9 |
| 53 | 63 | 11 | 63 |

Change 2 was done by running a script file fix20031204.sql.

# Change Log for 11/25/2003 CountyDB database

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20031125.EXE

This version, based on the previous 11/18/2003 version, includes the following changes and is intended to be used for Cimulus Task 29 and later.

**Change 1)** Added three new tables to the NMIM County database. These tables can be handled similarly to the County1999 and CountyBase tables, in that they are not used by NMIM and simply store data relevant to the database.

countymonthavg ‑ containing the multi‑year averages from Pechan.

countymonth2002 ‑ containing the 2002 temperatures from Pechan.

countymonth1999 ‑ containing the 1999 temperature values. (this’s also a backup of original CountyMonth table)

**Change 2)** Replaced the default values in the CountyMonth table with the values in the CountyMonth2002 table. That is, empty CountyMonth table, then load data from CountyMonth2002.

Changes 1 and 2 were done by running a script file fix20031125.sql.

**Change 3)** This script fixes the THC to VOC problem in the five original air toxic pollutants in SCCTOXICS table. It also fixes the zero values for exhEthGas, exhMTBEGas, and exhRFGGas for some dioxin pollutants.Change 3 was done by running a script file fixVOCTHC.sql.

**Change 4)** Dropped four NMIMOutput tables listed below, which belong to NMIMOutput database, not CountyDB.. Harvey M. pointed out that there’s a potential problem of making NMIM application users think that NIMIOutput and CountyDB database share the same database path/location, which may cause problem and NMIN application currently is not able to prevent users from doing so.

**NMINRun**

**NMIMVMTOutput**

**NMIMError**

**NMIMPollutantoutput**

# Change Log for 11/18/2003 CountyDB database

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20031118.EXE

This version, based on the previous 11/07/2003 version, includes the following changes and is intended to be used for Cimulus Task 29 and later.

**Change 1)** Corrected the fraction value of benz(a)anthracene for diesel in table SCCTOXICS. by running the following SQL statement:(91 records affected)

update scctoxics set exhdiesel = 0.00000071 where pollutantcodeid = 32

and (exhdiesel > 0.0000070 and exhdiesel < 0.0000072);

**Change 2)** Ran script to fix problems (a) There is an error in the value for the ETOHMktShare for GasolineID=3001 and GasolineID=3002. The current value is 100 and should be 1.00. And (b) Removed unused gasoline descriptions. There are 289 fuel descriptions in the Gasoline table which are never used. Although the database design allows for gasolines to be described but not used. (see C:\MySQL\data\CountyDB\readme\NMIMGasoline20031118.txt)

**Change Log for 11/07/2003 CountyDB database**

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20031107.EXE

This version, based on the previous 10/09/2003 version (see next page), includes the following changes and is intended to be used for Cimulus Task 29 and later.

**Change 1)** Corrected the dioxin emission factors from units of TEQ to g/mi and fixed the incorrect propionaldehyde emission factor in SCCTOXICS table. The fix was done by running the script file “fixPropAldehyde.sql” created by Ed Glover (Gwo made a little modification).

**Change Log for 10/09/2003 CountyDB database**

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\County20031009.EXE

For the purpose of easier to set up CountyDB/Output databases for running NMIM applications, all the **NMIM Output files** were included in this version. This version, based on the previous 7/24/2003 version (see next page), includes the following changes and is intended to be used for Cimulus Task 29 and later.

**Change 1)** Fixing data based on Pechan's “Comparison of NMIM County Database to NEI Modeling” final report dated 9/15/2003. The fix was done by running two scripts, pechanfix.sql and mainefix.sql.

**Change 2)** Added a new record into emissiontype table to include

emissiontypeid=124, emissiontype=All.(suggested by Harvey M.)

**Change 3)** Corrected PollutantCode, Pollutant\_Name in PollutantCodeID Table as suggested by Harvey M. by running the following code:

UPDATE PollutantCode SET PollutantCode = '16065831', PollutantName='Chromium (Cr3+)' WHERE

PollutantCode = '7440473' AND PollutantName='Chromim (Cr3+)';

UPDATE PollutantCode SET PollutantCode = '18540299', PollutantName='Chromium (Cr6+)' WHERE

PollutantCode = '18540299' AND PollutantName='Chromim (Cr6+)';

UPDATE PollutantCode SET PollutantCode = 'SOA', PollutantName='Secondary Organic Aerosol' WHERE

PollutantCode = 'SOA' AND PollutantName='Soluble Organic Aerosol';

UPDATE PollutantCode SET PollutantCode = '1746016', PollutantName='2,3,7,8‑Tetrachlorodibenzo‑p‑Dioxin' WHERE

PollutantCode = '600' AND PollutantName='2,3,7,8‑TCDD TEQ'**;**

**Change Log for 7/24/2003 CountyDB database**

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\Create\_db\County20030724.EXE

This version is based on the previous version dated 7/18/2003 (see next page).

**Change 1)** Run the following SQL code to update County database for the "Ethyl Benzene" issue reported by Megan:

UPDATE PollutantCode SET RunSpecPollutantIndex=21 WHERE

PollutantName='Ethyl Benzene';

**Change 2)** Delete the field “BenzExhDies” from table “SCC” by running the following SQL code:

alter table scc drop column BenzExhDies;

**Change Log for 7/18/2003 CountyDB database**

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\Create\_db\County20030718.EXE

**Note:** (1) This version works with the NMIM application of Cimulus Task 29. It **will not** work with any earlier versions of NMIM application due to the structure changes of NMIMOUTPUT database. (2) This version of CountyDB is based on 6/24/2003 CountyDB database design(see C:\MySQL\data\CountyDB\readme\countydb624.rtf for details)

There’re four changes made in this version of CountyDB:

**Change 1**) The three tables (as nine files), County, County1999, and Countybase should be copied into the CountyDB in order to provide a minimal number of representing counties for 1999 and 2002. These three tables are currently configured so that County and County1999 are identical. NMIM only uses County. Countybase contains the set of 241 NR representing counties developed by Mitch so that it will work for any year. County1999 has 106 representing counties and has been assured to be correct only for 1999 and 2002. If the user wants to do a run for 2010, she should copy Countybase into County. This operation requires copying three files:

Copy countybase.frm county.frm

Copy countybase.myd county.myd

Copy countybase.myi county.myi

To go back to the 1999 configuration, copy County1999 into County.

If there is any doubt, county1999 has an extra field called year, which is set to 1999 for all rows. Countybase does not have this extra field.

**Change 2)** Modified the NMIM OUTPUT design to reduce the NMIM output size: (see C:\MySQL\data\CountyDB\readme\Output624.rtf for details)

* Added a new key field "PollutantCodeID" (TINYINT(2)) to PollutantCode table. And changed the field "PollutantCode" to non-key.
* Added a new table "EmissionType".
* Added a new key field "SCCID" (SMALLINT) to SCC table. Changed the field "SCC" to non-key.
* Changed “RunId” to SMALLINT.
* Modified the FK/relationships affected by the above changes.

**Change 3)** All the external files/folders were deleted from the CountyDB folder, that is C:\MySQL\data\CountyDB (note that you may have a different folder name assigned by you during installing CountyDB).

**Change 4)** Further Changes made to the 5/12/2003 version of the ERG County Database

In addition to the serious problems which prevented the use of the database, there are smaller problems with the gasolines. The serious problems are addressed in a separate document. This document describes the changes to the gasolines.

1) California is aggressively phasing out MTBE in its fuels. This is reflected in the methodology that was to be used to generate future fuels for California. However, the adjustments were \*not\* applied to California gasolines. So, all of the California gasolines need to be regenerated using the correct adjustment profiles.

select year,month,fipsstateid,hwygasolineid

from countyyearmonth,gasoline

where fipsstateid=6

group by fipsstateid,hwygasolineid

order by year,month,hwygasolineid;

select \*

from gasoline

where gasolineid in

(171,172,173,1907,1908,1909,2193,2202,2205,2196,2208,2199,2194,2203,2206,2197,2209,2200,2195,2204,2207,2198,2210,2201);

The base year gasolines for California (171,172,173,1907,1908 and 1909) should be the same for calendar years 1999 through 2003. The adjustment factors for all of the California gasolines should be from factor set "W" from the spreadsheet "030425\_gasoline assignments and parameters.xls" provided by ERG for all calendar years after 2003. After proper adjustment, the adjusted fuels are substituted for the existing fuels in the Gasoline table.

delete

from gasoline

where gasolineid in

(2193,2202,2205,2196,2208,2199,2194,2203,2206,2197,2209,2200,2195,2204,2207,2198,2210,2201);

load data infile 'c:/mysql/data/california.csv'

into table gasoline

ignore 1 lines;

2) Puerto Rico and the Virgin Islands are assumed to have the same fuels as Hawaii. This is the same assumption used for the 1999 NEI inventory work. However, this is not very likely, since the mainland sources for Hawaii are likely to be quite different for Puerto Rico and the Virgin Islands. Instead, for NMIM, Puerto Rico and the Virgin Islands will be assumed to have the same fuels as Collier County, Florida (FIPS 12021).

create table florida

select \*

from countyyearmonth

where fipsstateid in (12) and

fipscountyid=21

limit 1000;

CREATE TABLE `newcym` (

`FIPSCountyId` smallint(3) unsigned NOT NULL default '0',

`Year` smallint(4) unsigned NOT NULL default '0',

`FIPSStateId` tinyint(2) unsigned NOT NULL default '0',

`Month` tinyint(2) unsigned NOT NULL default '0',

`HwyFuelDataSource` smallint(3) unsigned default NULL,

`HwyGasolineId` smallint(3) unsigned NOT NULL default '0',

`NRGasolineId` smallint(3) unsigned NOT NULL default '0',

`NRFuelDataSource` smallint(3) unsigned default NULL,

`HwyDieselId` tinyint(2) unsigned NOT NULL default '0',

`NRDieselId` tinyint(2) unsigned NOT NULL default '0',

`NGId` tinyint(2) unsigned NOT NULL default '0',

`HwyGasolineIdA` tinyint(2) unsigned NOT NULL default '0',

`HwyGasolineIdB` tinyint(2) unsigned NOT NULL default '0',

PRIMARY KEY (`FIPSCountyId`,`Year`,`FIPSStateId`,`Month`),

KEY `XIF22Count` (`NRFuelDataSource`),

KEY `XIF43CountyYearMonth` (`NRGasolineId`),

KEY `XIF10Count` (`HwyGasolineId`),

KEY `XIF44CountyYearMonth` (`NRDieselId`),

KEY `XIF8County` (`NGId`),

KEY `XIF5County` (`Year`,`FIPSCountyId`,`FIPSStateId`),

KEY `XIF46CountyYearMonth` (`HwyGasolineIdB`),

KEY `XIF26Count` (`HwyFuelDataSource`),

KEY `XIF6County` (`HwyDieselId`),

KEY `XIF45CountyYearMonth` (`HwyGasolineIdA`),

KEY `XIF4County` (`FIPSCountyId`,`FIPSStateId`,`Month`)

) TYPE=MyISAM;

insert into newcym (fipsstateid,fipscountyid,year,month,ngid,hwydieselid,

hwygasolineid,nrgasolineid,nrdieselid,hwyfueldatasource,nrfueldatasource,

hwygasolineida,hwygasolineidb)

select a.fipsstateid as fipsstateid,

a.fipscountyid as fipscountyid,

a.year as year,

a.month as month,

if(a.fipsstateid in (72,78),b.ngid,a.ngid) as ngid,

if(a.fipsstateid in (72,78),b.hwydieselid,a.hwydieselid) as hwydieselid,

if(a.fipsstateid in (72,78),b.hwygasolineid,a.hwygasolineid) as hwygasolineid,

if(a.fipsstateid in (72,78),b.nrgasolineid,a.nrgasolineid) as nrgasolineid,

if(a.fipsstateid in (72,78),b.nrdieselid,a.nrdieselid) as nrdieselid,

if(a.fipsstateid in (72,78),b.hwyfueldatasource,a.hwyfueldatasource) as hwyfueldatasource,

if(a.fipsstateid in (72,78),b.nrfueldatasource,a.nrfueldatasource) as nrfueldatasource,

if(a.fipsstateid in (72,78),b.hwygasolineida,a.hwygasolineida) as hwygasolineida,

if(a.fipsstateid in (72,78),b.hwygasolineidb,a.hwygasolineidb) as hwygasolineidb

from countyyearmonth as a, florida as b

where

a.year = b.year and

a.month = b.month

limit 5000000;

drop table florida;

drop table countyyearmonth;

rename table newcym to countyyearmonth;

~~create table newcym~~

~~select a.fipsstateid,a.fipscountyid,a.year,a.month,~~

~~if(a.fipsstateid in (72,78),b.ngid,a.ngid) as ngid,~~

~~if(a.fipsstateid in (72,78),b.hwydieselid,a.hwydieselid) as hwydieselid,~~

~~if(a.fipsstateid in (72,78),b.hwygasolineid,a.hwygasolineid) as hwygasolineid,~~

~~if(a.fipsstateid in (72,78),b.nrgasolineid,a.nrgasolineid) as nrgasolineid,~~

~~if(a.fipsstateid in (72,78),b.nrdieselid,a.nrdieselid) as nrdieselid,~~

~~if(a.fipsstateid in (72,78),b.hwyfueldatasource,a.hwyfueldatasource) as hwyfueldatasource,~~

~~if(a.fipsstateid in (72,78),b.nrfueldatasource,a.nrfueldatasource) as nrfueldatasource,~~

~~if(a.fipsstateid in (72,78),b.hwygasolineida,a.hwygasolineida) as hwygasolineida,~~

~~if(a.fipsstateid in (72,78),b.hwygasolineidb,a.hwygasolineidb) as hwygasolineidb~~

~~from countyyearmonth as a, florida as b~~

~~where~~

~~a.year = b.year and~~

~~a.month = b.month~~

~~limit 5000000;~~

~~drop table florida;~~

~~drop table countyyearmonth;~~

rename table newcym to countyyearmonth;

3) I'm not sure this is documented anywhere, but the gasoline maximum sulfur level for all gasolines is 303 ppm in 2004 and 2005, 87 ppm in 2006 and 2007, and 80 in 2008 and later calendar years in all counties of all states.

**Change Log for 6/20/2003 CountyDB database**

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\Create\_db\County20030620.EXE

Added a new column, RunSpecPollutantIndex TINY (2) NOT NULL , to the PollutantCode table in the CountyDB. This change applies to Task 26 and later. There three files involved in this change, namely pollutantcode.frm, pollutantcode.myd, and pollutantcode.myi.

Attached below is the new PollutantCode table as of 6/20/2003 from Cimulus.

|  |  |  |
| --- | --- | --- |
| **PollutantCodeID** | **PollutantName** | **RunSpecPollutantIndex** |
| 100414 | Ethyl Benzene | 17 |
| 100425 | Styrene | 21 |
| 106990 | 1,3-Butadiene | 18 |
| 107028 | Acrolein | 16 |
| 108883 | Toluene | 21 |
| 110543 | Hexane | 21 |
| 120127 | Anthracene | 21 |
| 123386 | Propionaldehyde | 21 |
| 129000 | Pyrene | 21 |
| 1330207 | Xylene | 21 |
| 1634044 | MTBE | 20 |
| 18540299 | Chromim (Cr6+) | 21 |
| 191242 | Benzo(g,h,i)perylene | 21 |
| 193395 | Indeno(1,2,3,c,d)pyrene | 21 |
| 19408743 | 1,2,3,7,8,9-Hexachlorodibenzo-p-Dioxin | 22 |
| 205992 | Benzo(b)fluoranthene | 21 |
| 206440 | Fluoranthene | 21 |
| 207089 | Benzo(k)fluoranthene | 21 |
| 208968 | Acenaphthylene | 21 |
| 218019 | Chrysene | 21 |
| 3268879 | Octachlorodibenzo-p-dioxin | 22 |
| 35822469 | 1,2,3,4,6,7,8-Heptachlorodibenzo-p-Dioxin | 22 |
| 39001020 | Octachlorodibenzofuran | 22 |
| 39227286 | 1,2,3,4,7,8-Hexachlorodibenzo-p-Dioxin | 22 |
| 40321764 | 1,2,3,7,8-Pentachlorodibenzo-p-Dioxin | 22 |
| 50000 | Formaldehyde | 19 |
| 50328 | Benzo(a)pyrene | 21 |
| 51207319 | 2,3,7,8-Tetrachlorodibenzofuran | 22 |
| 53703 | Dibenzo(a,h)anthracene | 21 |
| 540841 | 2,2,4-Trimethylpentane | 21 |
| 55673897 | 1,2,3,4,7,8,9-Heptachlorodibenzofuran | 22 |
| 56553 | Benz(a)anthracene | 21 |
| 57117314 | 2,3,4,7,8-Pentachlorodibenzofuran | 22 |
| 57117416 | 1,2,3,7,8-Pentachlorodibenzofuran | 22 |
| 57117449 | 1,2,3,6,7,8-Hexachlorodibenzofuran | 22 |
| 57653857 | 1,2,3,6,7,8-Hexachlorodibenzo-p-Dioxin | 22 |
| 600 | 2,3,7,8-TCDD TEQ | 22 |
| 60851345 | 2,3,4,6,7,8-Hexachlorodibenzofuran | 22 |
| 67562394 | 1,2,3,4,6,7,8-Heptachlorodibenzofuran | 22 |
| 70648269 | 1,2,3,4,7,8-Hexachlorodibenzofuran | 22 |
| 71432 | Benzene | 17 |
| 72918219 | 1,2,3,7,8,9-Hexachlorodibenzofuran | 22 |
| 7439965 | Manganese | 21 |
| 7439976 | Mercury | 21 |
| 7440020 | Nickel | 21 |
| 7440382 | Arsenic | 21 |
| 7440473 | Chromim (Cr3+) | 21 |
| 75070 | Acetaldehyde | 15 |
| 83329 | Acenaphthene | 21 |
| 85018 | Phenanthrene | 21 |
| 86737 | Fluorene | 21 |
| 91203 | Naphthalene | 21 |
| CH4 | Methane | 21 |
| CO | Carbon Monoxide | 9 |
| CO2 | Carbon Dioxide | 10 |
| HC | Total Hydrocarbons | 11 |
| NH3 | Ammonia | 12 |
| NMHC | Non-Methane Hydrocarbons | 11 |
| NMOG | Non-Methane Organic Gases | 11 |
| NOx | Nitrogen Oxides | 13 |
| PM10-PRI | Primary PM10 (Filterables and Condensibles) | 100 |
| PM25-PRI | Primary PM2.5 (Filterables and Condensibles) | 100 |
| SO2 | Sulfur Dioxide | 14 |
| SOA | Soluble Organic Aerosol | 8 |
| TOG | Total Organic Gases | 11 |
| VOC | Volatile Organic Compounds | 11 |

**Change Log for 6/17/2003 CountyDB database**

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\db\_history\County20030617.EXE

**This version of CountyDB dated 6/17/2003 was modified and to be used by Task 26 and after. The SCC table has been changed by deleting fields as follows:**

**alter table scc drop column BenzExhGas;**

**alter table scc drop column BenzExhEth;**

**alter table scc drop column BenzExhMTBE;**

**alter table scc drop column BenzExhRFG;**

**alter table scc drop column BenzEvapGas;**

**alter table scc drop column BenzEvapEth;**

**alter table scc drop column BenzEvapMTBE;**

**alter table scc drop column BenzEvapRFG;**

**alter table scc drop column ButaExhGas;**

**alter table scc drop column ButaExhEth;**

**alter table scc drop column ButaExhMTBE;**

**alter table scc drop column ButaExhRFG;**

**alter table scc drop column FormExhGas;**

**alter table scc drop column FormExhEth;**

**alter table scc drop column FormExhMTBE;**

**alter table scc drop column FormExhRFG;**

**alter table scc drop column AcetExhGas;**

**alter table scc drop column AcetExhEth;**

**alter table scc drop column AcetExhMTBE;**

**alter table scc drop column AcetExhRFG;**

**alter table scc drop column AcroExhGas;**

**alter table scc drop column AcroExhEth;**

**alter table scc drop column AcroExhMTBE;**

**alter table scc drop column AcroExhRFG;**

**alter table scc drop column MTBEExhGas;**

**alter table scc drop column MTBEExhEth;**

**alter table scc drop column MTBEExhMTBE;**

**alter table scc drop column MTBEExhRFG;**

**alter table scc drop column MTBEEvapGas;**

**alter table scc drop column MTBEEvapEth;**

**alter table scc drop column MTBEEvapMTBE;**

**alter table scc drop column MTBEEvapRFG;**

**alter table scc drop column ButaExhDies;**

**alter table scc drop column FormExhDies;**

**alter table scc drop column AcetExhDies;**

**alter table scc drop column AcroExhDies;**

**Change Log for 6/12/2003 CountyDB database**

**Production Database:** H:\AMD\MOBILE\NMIM\CountyDB\db\_history\County20030612.EXE

This version of CountyDB dated 6/12/2003 was created based on Dave B.’s 6/16/2003 County Database and is intended to be used by NMIM applications that are the deliverables of Cimulus Task 25 and earlier.

Deleted unwanted tables and their myd and myi files:

* cymmtbephsout.frm, cymmtbephsout.myd, and cymmtbephsout.myi.
* gasmtbephsout.frm, gasmtbephsout.myd, and gasmtbephsout.myi.
* gas2mtbephsout.frm, gas2mtbephsout.myd, and gas2mtbephsout.myi.
* imfilenames.frm, imfilenames.myd, and imfilenames.myi.
* mainegas.frm, mainegas.myd, and mainegas.myi.

Added three new tables and populated data: pollutantcode, scctoxics, and scc.

Modified CountyDB database:

* M6VClass table: added two fields “Vtype TinyInt(2) Unsigned Not NULL” and “p5class TinyInt(2) Unsigned Not NULL” and populated data.
* County table: Added 4 new fields required by Task 28:
* OzoneSeasonStartMonth TINYINT(2), NOT NULL, all records have value 6.
* OzoneSeasonStartDay TINYINT(2), NOT NULL, all records have value 1.
* OzoneSeasonEndMonth TINYINT(2), NOT NULL, all records have value 8.
* OzoneSeasonEndDay TINYINT(2), NOT NULL, all records have value 31.

Modified Erwin CountDB design:

* Changed data length to (6,4) for the following numeric fields in gasoline table: etohmktshare, mtbemktshare, tamemktshare, and totethermktshare.
* Changed data length to (6,4) for field mktshare.
* Added 4 new fields into County table:
* OzoneSeasonStartMonth TINYINT(2), NOT NULL.
* OzoneSeasonStartDay TINYINT(2), NOT NULL.
* OzoneSeasonEndMonth TINYINT(2), NOT NULL.
* OzoneSeasonEndDay TINYINT(2), NOT NULL.

Data corrected:

* Changed the value of mtbemktshare for gasoline id=188 from 68.92 to 0.6892.
* Set CountyYear.Stage2Pct = 0 for all records. That is , changes values of 95 to 0. (this from team’s 6/9/2003 meeting conclusion)
* SCC table: deleted two records with ids = 2268008005 and 2260008005.

Changes made to the 5/12/2003 version of the ERG County Database are:

(Created 6/6/2003 by Dave B., Last updated 6/12/2003 by Gwo S.)

1) Set RVPOxyWaiver=1 in Gasoline for all cases. This will force MOBILE6 to always use the RVP value stored for that fuel regardless of the ETOH oxygenated volume percent. Otherwise, MOBILE6 will adjust the RVP, assuming splash blending of the alcohol, whenever the ETOH volume percent is not zero.

UPDATE gasoline SET rvpoxywaiver=1;

2) The oxygenate market share for GasolineID=188 is greater than one (131.08%). This fuel is only used in 89 counties in Indiana. This fuel is only used in calendar years 1999 (base year) through 2003. Only 9 fuels are ever assigned to counties in Indiana in any calendar year (48, 49, 50, 188, 189, 190, 297, 298 and 299). Fuel 188 has a ETOHMktShare of 31.08%, which is exactly the same ETOHMktShare in fuels 48, 49, 50, 189 and 190. The other fuels in Indiana all have 100% ETOHMktShare and zero MTBEMktShare. All fuels used by counties that use fuel 188 have a MTBEMktShare of 68.92% for all other fuels used in other calendar years. Also, fuel 188 is the combination of two fuels, 68 and 69, from the Gasoline2 table. The MTBE fuel (69) has a 68.92% market share.

This leads me to believe that the MTBEMktShare of 100% for GasolineID=188 is an isolated error. I conclude that this fuel should have the same MTBEMktShare as the other fuels in Indiana which share the same ETOHMktShare. I have set the MTBEMktShare for GasolineID=188 to be 68.92%.

UPDATE gasoline SET MTBEMktShare=68.92 WHERE GasolineID=188;

1. Some values for MTBEVolume exceed the MOBILE6 maximum value of 15.1%. This problem only affects future (not 1999 base year) gasoline. This is due to the generic multiplicative adjustment of MTBE volumes in future calendar years. The MTBE volumes in some counties of Nevada, Maine, Hawaii, Puerto Rico and the Virgin Islands are too large. This is not an error in applying the EPA supplied methodology. However, these results were not expected and must be changed in order for NMIM to use the data.

In Nevada, the MTBE volume adjustment for 2007 and later calendar years (5.18) is so large that some counties with significant MTBE volumes in the base year end up with extreme MTBE volumes in 2007.

Only five counties (7,11,13,15 and 33) are affected. Only six gasolines (1510, 1511, 1512, 1514, 1515 and 1516) have extreme MTBE volume values. The MTBEvolumes for these six gasolines were capped at the MOBILE6 maximum (15.1%). Three additional gasolines with high MTBE volumes (2103, 2104 and 2105) are never used and were deleted.

update gasoline

set mtbevolume=15.1

where gasolineid in (1510,1511,1512,1514,1515,1516);

delete from gasoline

where gasolineid in (2103,2104,2105);

In Maine, the RFG program was eliminated early in 1999, which listed Maine as a "non-RFG" area for adjustments. However, fuels delivered in 1999 to some counties still contained large amounts of MTBE, likely due to fuel already contracted before the RFG requirement was lifted. So, rather than continuing to receive MTBE containing fuel, Maine should get less MTBE in future calendar years in those counties.

There are only two sets of gasoline used in Maine counties, low and high MTBE. The solution is to set the HwyGasolineID (and NRGasolineID) in all 2000 and later calendar years in the high MTBE counties to the same HwyGasolineID values used in other Maine counties. FIPSCountyID=1 is representative of high MTBE counties and FIPSCountyID=3 is representative of low MTBE counties. Match the gasolines by year and month in these two counties and replace the HwyGasolineID and NRGasolineID values for the high MTBE counties with the corresponding low MTBE HwyGasolineID values in all calendar years after 1999.

CREATE TABLE MaineGas

SELECT a.hwygasolineid as baseid,b.hwygasolineid as newid

FROM countyyearmonth as a, countyyearmonth as b

WHERE

a.fipsstateid=b.fipsstateid AND

a.year=b.year AND

a.month=b.month AND

a.fipsstateid=23 AND

a.fipscountyid=1 AND

b.fipscountyid=3

GROUP BY a.hwygasolineid;

CREATE TABLE `cymnew` (

`FIPSCountyId` smallint(3) unsigned NOT NULL default '0',

`Year` smallint(4) unsigned NOT NULL default '0',

`FIPSStateId` tinyint(2) unsigned NOT NULL default '0',

`Month` tinyint(2) unsigned NOT NULL default '0',

`HwyFuelDataSource` smallint(3) unsigned default NULL,

`HwyGasolineId` smallint(3) unsigned NOT NULL default '0',

`NRGasolineId` smallint(3) unsigned NOT NULL default '0',

`NRFuelDataSource` smallint(3) unsigned default NULL,

`HwyDieselId` tinyint(2) unsigned NOT NULL default '0',

`NRDieselId` tinyint(2) unsigned NOT NULL default '0',

`NGId` tinyint(2) unsigned NOT NULL default '0',

`HwyGasolineIdA` tinyint(2) unsigned NOT NULL default '0',

`HwyGasolineIdB` tinyint(2) unsigned NOT NULL default '0',

PRIMARY KEY (`FIPSCountyId`,`Year`,`FIPSStateId`,`Month`),

KEY `XIF22Count` (`NRFuelDataSource`),

KEY `XIF43CountyYearMonth` (`NRGasolineId`),

KEY `XIF10Count` (`HwyGasolineId`),

KEY `XIF44CountyYearMonth` (`NRDieselId`),

KEY `XIF8County` (`NGId`),

KEY `XIF5County` (`Year`,`FIPSCountyId`,`FIPSStateId`),

KEY `XIF46CountyYearMonth` (`HwyGasolineIdB`),

KEY `XIF26Count` (`HwyFuelDataSource`),

KEY `XIF6County` (`HwyDieselId`),

KEY `XIF45CountyYearMonth` (`HwyGasolineIdA`),

KEY `XIF4County` (`FIPSCountyId`,`FIPSStateId`,`Month`)

) TYPE=MyISAM;

Insert into cymnew (fipsstateid,fipscountyid,year,month,ngid,hwydieselid,hwygasolineid,nrgasolineid,nrdieselid,hwyfueldatasource,nrfueldatasource,hwygasolineida,hwygasolineidb)

select fipsstateid,fipscountyid,year,month,ngid,hwydieselid,

if(isnull(baseid),hwygasolineid,if(year>1999,newid,hwygasolineid)) as hwygasolineid,

if(isnull(baseid),nrgasolineid,if(year>1999,newid,nrgasolineid)) as nrgasolineid,

nrdieselid,hwyfueldatasource,nrfueldatasource,hwygasolineida,hwygasolineidb

from countyyearmonth left join mainegas

on hwygasoliineid=baseid;

drop countyyearmonth;

rename table cymnew to countyyearmonth;

delete from gasoline

where gasolineid in (423,424,425,938,939,940,941,942,943,944,945,946);

In addition to these serious problems we have identified problems which, although serious, do not prevent the use of the database. These problems, and changes in the fuel assumptions since the delivery of the database, will require a much more expansive recalculation of gasoline parameters than can be attempted now. So, the following gasoline problems were \*not\* fixed in the database:

California is aggressively phasing out MTBE in its fuels. This is reflected in the methodology that was to be used to generate future fuels for California. However, the adjustments were \*not\* applied to California gasolines. So, all of the California gasolines need to be regenerated using the correct adjustment profiles (V and W).

Hawaii gets most of it's gasoline from the mainland (California). In 1999, Hawaiian gasoline has large amounts of MTBE. However, California is aggressively phasing out MTBE. Therefore, Hawaii should use the same future year profiles as California (V and W).

Puerto Rico and the Virgin Islands are assumed to have the same fuels as Hawaii. This is not very likely, since the mainland sources for Hawaii are likely to be geographically quite different than for Puerto Rico and the Virgin Islands.

1. There are 21 gasoline oxygenate volume sums that are slightly greater than 1.0. This may cause problems in MOBILE6. In all cases the MTBE volume is the largest fraction. The MTBEVolume in all cases was reduced so that the sum of the oxygenate volumes was equal to 100. Some sums are slightly less than 100. These values were not changed.

UPDATE gasoline

SET mtbemktshare =

mtbemktshare‑(etohmktshare+mtbemktshare+etbemktshare+tamemktshare‑100)

WHERE etohmktshare+mtbemktshare+etbemktshare+tamemktshare > 100;

UPDATE gasoline

SET mtbemktshare=mtbemktshare+0.01

WHERE (etohmktshare+mtbemktshare+etbemktshare+tamemktshare) > 99 and

(etohmktshare+mtbemktshare+etbemktshare+tamemktshare) < 99.999;

1. All of the 1858 records in Gasoline and all 81 records in Gasoline2 have market share values in percent format, rather than the fraction format required by MOBILE6. In order to retain the number of significant digits in the values, the market share fields must be modified before dividing the values by 100.

ALTER TABLE gasoline

MODIFY etohmktshare decimal(6,4) not null,

MODIFY mtbemktshare decimal(6,4) not null,

MODIFY etbemktshare decimal(6,4) not null,

MODIFY tamemktshare decimal(6,4) not null,

MODIFY totethermktshare decimal(6,4) not null;

The CountyDB Erwin design also has been modified to include the above data-length changes of market shares. See H:\AMD\MOBILE\NMIM\CountyDB\CountyDB20030612.ER1.

UPDATE gasoline

SET etohmktshare = etohmktshare/100,

mtbemktshare = mtbemktshare/100,

etbemktshare = etbemktshare/100,

tamemktshare = tamemktshare/100,

totethermktshare=totethermktshare/100;

ALTER TABLE gasoline2

MODIFY mktshare decimal(6,4) not null;

The CountyDB Erwin design also has been modified to include the above data-length change of market share. See H:\AMD\MOBILE\NMIM\CountyDB\CountyDB20030612.ER1.

UPDATE gasoline2

SET mktshare = mktshare/100;

1. Some counties have representing counties that do not match their basic parameters (as they should). Rather than investigate to determine what the appropriate representing county should be, each of the mis-matched counties will be set to represent itself. This will significantly increase the number of representing counties. If NMIM performance is a problem, it can be improved by investigating these counties to determine their appropriate representing county.

A. Check that counties within groups, both highway and nonroad, do not contain conflicting values, identifying any counties that have such problems.

B. Modify the county groupings to correct any such problems discovered by making more counties represent themselves.

On the current NMIM county database this process roughly doubles the number of county groups.

The script files are meant to be run in sequence as follows:

1. checkhwygroups.sql and/or checknrgroups.sql carry out function A above.

2. makegroupstofix.sql constructs consolidated, but not completely unique, lists of counties with problems.

3. makenewcountytable.sql constructs new County table. This is the only script file which modifies the database.

4. step 1 can be repeated if desired at this point to confirm that the database no longer has problems.

5. cleanupaftercountygroupfix.sql deletes all files created in the process that do not belong in county database.

#checkhwygroups.sql

/\* MySQL script file to check that counties in the same highway group have

same information \*/

USE CountyDB;

/\* Look for Counties in the same highway group with different altitude values \*/

DROP TABLE IF EXISTS hwyaltitudeprob;

CREATE TABLE hwyaltitudeprob

SELECT C.FIPSStateID, C.FIPSCountyID, C.altitude,

C.HWYRepFIPSCntyID, C2.altitude AS REPaltitude

FROM County AS C, County AS C2

WHERE C.FIPSStateID = C2.FIPSStateID AND C2.FIPSCountyID = C.HWYRepFIPSCntyID

AND C2.altitude <> C.altitude;

/\* Look for Counties in the same highway group with different IMFileName values \*/

DROP TABLE IF EXISTS imfileprob;

CREATE TABLE imfileprob

SELECT C.FIPSStateID, C.FIPSCountyID, CY1.year, CY1.IMFileName,

C.HWYRepFIPSCntyID, CY2.IMFileName AS REPIMFileName

FROM County AS C, CountyYear AS CY1, CountyYear AS CY2

WHERE C.FIPSStateID = CY1.FIPSStateID AND C.FIPSCountyID = CY1.FIPSCountyID

AND C.FIPSStateID = CY2.FIPSStateID AND C.HWYRepFIPSCntyID = CY2.FIPSCountyID

AND CY1.year = CY2.year

AND CY1.IMFileName <> CY2.IMFileName;

/\* Look for Counties in the same highway group with different ATPFileName values \*/

DROP TABLE IF EXISTS atpfileprob;

CREATE TABLE atpfileprob

SELECT C.FIPSStateID, C.FIPSCountyID, CY1.year, CY1.ATPFileName,

C.HWYRepFIPSCntyID, CY2.ATPFileName AS REPATPFileName

FROM County AS C, CountyYear AS CY1, CountyYear AS CY2

WHERE C.FIPSStateID = CY1.FIPSStateID AND C.FIPSCountyID = CY1.FIPSCountyID

AND C.FIPSStateID = CY2.FIPSStateID AND C.HWYRepFIPSCntyID = CY2.FIPSCountyID

AND CY1.year = CY2.year

AND CY1.ATPFileName <> CY2.ATPFileName;

/\* Look for Counties in the same highway group with different maxTemp values \*/

DROP TABLE IF EXISTS maxtempprob;

CREATE TABLE maxtempprob

SELECT C.FIPSStateID, C.FIPSCountyID, CM1.month, CM1.maxTemp,

C.HwyRepFIPSCntyID, CM2.maxTemp AS REPMaxTemp

FROM County AS C, CountyMonth AS CM1, CountyMonth AS CM2

WHERE C.FIPSStateID = CM1.FIPSStateID AND C.FIPSCountyID = CM1.FIPSCountyID

AND C.FIPSStateID = CM2.FIPSStateID AND C.HwyRepFIPSCntyID = CM2.FIPSCountyID

AND CM1.month = CM2.month

AND CM1.maxTemp <> CM2.maxTemp;

/\* Look for Counties in the same highway group with different minTemp values \*/

DROP TABLE IF EXISTS mintempprob;

CREATE TABLE mintempprob

SELECT C.FIPSStateID, C.FIPSCountyID, CM1.month, CM1.minTemp,

C.HwyRepFIPSCntyID, CM2.minTemp AS REPminTemp

FROM County AS C, CountyMonth AS CM1, CountyMonth AS CM2

WHERE C.FIPSStateID = CM1.FIPSStateID AND C.FIPSCountyID = CM1.FIPSCountyID

AND C.FIPSStateID = CM2.FIPSStateID AND C.HwyRepFIPSCntyID = CM2.FIPSCountyID

AND CM1.month = CM2.month

AND CM1.minTemp <> CM2.minTemp;

/\* Look for Counties in the same highway group with different highway gasolineID values \*/

DROP TABLE IF EXISTS hwygasprob;

CREATE TABLE hwygasprob

SELECT C.FIPSStateID, C.FIPSCountyID, CYM1.year, CYM1.month, CYM1.HwyGasolineID,

C.HWYRepFIPSCntyID, CYM2.HwyGasolineID AS REPHwyGasolineID

FROM County AS C, CountyYearMonth AS CYM1, CountyYearMonth AS CYM2

WHERE C.FIPSStateID = CYM1.FIPSStateID AND C.FIPSCountyID = CYM1.FIPSCountyID

AND C.FIPSStateID = CYM2.FIPSStateID AND C.HWYRepFIPSCntyID = CYM2.FIPSCountyID

AND CYM1.year = CYM2.year AND CYM1.month = CYM2.month

AND CYM1.HwyGasolineID <> CYM2.HwyGasolineID;

#checknrgroups.sql

/\* MySQL script file to check that counties in the same nonroad group have

same information \*/

USE CountyDB;

/\* Look for Counties in the same nonroad group with different altitude values \*/

DROP TABLE IF EXISTS nraltitudeprob;

CREATE TABLE nraltitudeprob

SELECT C.FIPSStateID, C.FIPSCountyID, C.altitude,

C.NRRepFIPSCntyID, C2.altitude AS REPaltitude

FROM County AS C, County AS C2

WHERE C.FIPSStateID = C2.FIPSStateID AND C2.FIPSCountyID = C.NRRepFIPSCntyID

AND C2.altitude <> C.altitude;

/\* Look for Counties in the same nonroad group with different avgTemp values \*/

DROP TABLE IF EXISTS avgtempprob;

CREATE TABLE avgtempprob

SELECT C.FIPSStateID, C.FIPSCountyID, CM1.month, CM1.avgTemp,

C.NRRepFIPSCntyID, CM2.avgTemp AS REPAvgTemp

FROM County AS C, CountyMonth AS CM1, CountyMonth AS CM2

WHERE C.FIPSStateID = CM1.FIPSStateID AND C.FIPSCountyID = CM1.FIPSCountyID

AND C.FIPSStateID = CM2.FIPSStateID AND C.NRRepFIPSCntyID = CM2.FIPSCountyID

AND CM1.month = CM2.month

AND CM1.avgTemp <> CM2.avgTemp;

/\* Look for Counties in the same nonroad group with different nonroad gasolineID values \*/

DROP TABLE IF EXISTS nrgasprob;

CREATE TABLE nrgasprob

SELECT C.FIPSStateID, C.FIPSCountyID, CYM1.year, CYM1.month, CYM1.NRGasolineID,

C.NRRepFIPSCntyID, CYM2.NRGasolineID AS REPNRGasolineID

FROM County AS C, CountyYearMonth AS CYM1, CountyYearMonth AS CYM2

WHERE C.FIPSStateID = CYM1.FIPSStateID AND C.FIPSCountyID = CYM1.FIPSCountyID

AND C.FIPSStateID = CYM2.FIPSStateID AND C.NRRepFIPSCntyID = CYM2.FIPSCountyID

AND CYM1.year = CYM2.year AND CYM1.month = CYM2.month

AND CYM1.NRGasolineID <> CYM2.NRGasolineID;

#makegroupstofix.sql

/\* MySQL script file to prepare two lists of NMIM counties

\* (one for highway, one for nonroad) identified to

\* become their own County Group \*/

USE CountyDB;

DROP TABLE IF EXISTS HwyCountyToFix;

CREATE TABLE HwyCountyToFix (

FIPSStateID SMALLINT,

FIPSCountyID SMALLINT

);

INSERT INTO HwyCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM hwyaltitudeprob;

INSERT INTO HwyCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM imfileprob;

INSERT INTO HwyCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM atpfileprob;

INSERT INTO HwyCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM maxtempprob;

INSERT INTO HwyCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM mintempprob;

INSERT INTO HwyCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM hwygasprob;

DROP TABLE IF EXISTS NRCountyToFix;

CREATE TABLE NRCountyToFix (

FIPSStateID SMALLINT,

FIPSCountyID SMALLINT

);

INSERT INTO NRCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM nraltitudeprob;

INSERT INTO NRCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM avgtempprob;

INSERT INTO NRCountyToFix

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM nrgasprob;

#makenewcountytable.sql

/\* MySQL script file to update NMIM County table to make more

counties represent themselves (where they have different data). Assumes

that checkhwygroups.sql, checknrgroups.sql, and makegroupstofix.sql

have been run. \*/

USE CountyDB;

/\* First, eliminate duplicates FROM LISTS of counties to fix. \*/

DROP TABLE IF EXISTS HwyCountyToFix2;

CREATE TEMPORARY TABLE HwyCountyToFix2

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM HwyCountyToFix;

DROP TABLE IF EXISTS NRCountyToFix2;

CREATE TEMPORARY TABLE NRCountyToFix2

SELECT DISTINCT FIPSStateID, FIPSCountyID FROM NRCountyToFix;

/\* Second, make new copy of County table

with HwyRepCountyID changed where necessary \*/

DROP TABLE IF EXISTS County2;

CREATE TABLE County2

SELECT C.FIPSStateID, C.FIPSCountyID, C.Altitude,

C.CountyName, C.NRRepFIPSCntyID,

IF(ISNULL(Hwy.FIPSCountyID),C.HwyRepFIPSCntyID, C.FIPSCountyID) AS HwyRepFIPSCntyID

FROM County AS C

LEFT JOIN HwyCountyToFix2 AS Hwy USING (FIPSStateID, FIPSCountyID);

/\* Now repopulate County Table from County2 with

NRRepCountyID changed where necessary \*/

TRUNCATE TABLE County;

INSERT INTO County (FIPSStateID, FIPSCountyID, Altitude,

CountyName, NRRepFIPSCntyID, HwyRepFIPSCntyID)

SELECT C.FIPSStateID, C.FIPSCountyID, C.Altitude, C.CountyName,

IF(ISNULL(NR.FIPSCountyID),C.NRRepFIPSCntyID, C.FIPSCountyID),

C.HwyRepFipsCntyID

FROM County2 AS C

LEFT JOIN NRCountyToFix2 AS NR USING (FIPSStateID, FIPSCountyID);

#cleanupaftercountygroupfix.sql

/\* MySQL script file to cleanup extraneaous tables created by

checkhwygroups.sql, checknrgroups.sql, makegroupstofix.sql, and

makenewcountytable.sql \*/

USE CountyDB;

DROP TABLE IF EXISTS hwyaltitudeprob;

DROP TABLE IF EXISTS imfileprob;

DROP TABLE IF EXISTS atpfileprob;

DROP TABLE IF EXISTS maxtempprob;

DROP TABLE IF EXISTS mintempprob;

DROP TABLE IF EXISTS hwygasprob;

DROP TABLE IF EXISTS nraltitudeprob;

DROP TABLE IF EXISTS avgtempprob;

DROP TABLE IF EXISTS nrgasprob;

DROP TABLE IF EXISTS hwycountytofix;

DROP TABLE IF EXISTS nrcountytofix;

DROP TABLE IF EXISTS county2;

NCD20051207 database still has states with stage2pct values non zero.

mysql> select distinct fipsstateid, stage2pct from countyyear

‑> where stage2pct !=0;

+‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑+

| fipsstateid | stage2pct |

+‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑+

| 51 | 77 |

| 24 | 100 |

| 11 | 90 |

+‑‑‑‑‑‑‑‑‑‑‑‑‑+‑‑‑‑‑‑‑‑‑‑‑+

3 rows in set (0.63 sec)

This should be fixed to 0%.

mysql> update countyyear set stage2pct=0;

Query OK, 653 rows affected (0.09 sec)

Rows matched: 167594 Changed: 653 Warnings: 0

mysql> select distinct fipsstateid, stage2pct from countyyear

‑> where stage2pct !=0;

Empty set (0.06 sec)

ExternalFiles

File 19000.sea replaced with new version.

ChangeLog **for NCD20060725**

**1**. External Files: NLEV files updated to begin Tier 2 standards in 2004. djb 20051214

This is now a standard external data files. Files updated are:

10000.nlv, 11001.nlv, 23000.nlv, 24000.nlv, 25000.nlv, 36000.nlv, 42000.nlv,

44000.nlv, 50000.nlv, 51000.nlv. (former files: Maimp.d, Meimp.d, Nimpotr.d,nyimp.d, P94imp.d, vtimp.d).

Batch file to Update County database with corrected .nlv External files.

copy T:\NMIMRuns\MSAT\NLEVFiles\Maimp.d C:\mysql\data\NCD20060705\ExternalFiles\25000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\Meimp.d C:\mysql\data\NCD20060705\ExternalFiles\23000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\vtimp.d C:\mysql\data\NCD20060705\ExternalFiles\50000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nyimp.d C:\mysql\data\NCD20060705\ExternalFiles\36000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060705\ExternalFiles\42000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060705\ExternalFiles\10000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060705\ExternalFiles\11001.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060705\ExternalFiles\24000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060705\ExternalFiles\44000.nlv /a /y ;

copy T:\NMIMRuns\MSAT\NLEVFiles\nimpotr.d C:\mysql\data\NCD20060705\ExternalFiles\51000.nlv /a /y ;

**2.** The SCCToxics table has evaporative ratios for Dies, LPG, and CNG, and it shouldn't.

Script to set all evap emission factors in the SCCToxics table to NULL for all cases where fueltype is dies, lpg, and cng.:

update scctoxics t, scc s

set

t.evapEthGas = NULL,

t.evapBaseGas = NULL,

t.evapMTBEGas = NULL,

t.evapRFGGas = NULL

where s.fuelType in ('Dies','LPG','CNG') and s.sccid = t.sccid;

Script to set the RatioType to PM10 for naphthalene for all diesel SCCs. (It was incorrectly set to PMVOC for onroad diesel).

update scctoxics t, scc s

set t.ratioType='PM10'

where t.pollutantCodeId=52

and s.fuelType='Dies' and t.sccid=s.sccid;