Software Technical Documentation

For Software to Estimate Future Activity and Air Emissions from Electric Generating Units (EGUs)

Prepared by:

AMEC Environment & Infrastructure, Inc.
4021 Stirrup Creek, Suite 100,
Durham, NC 27703

Prepared for:

Mid-Atlantic Regional Air Management Association, Inc.
8600 Lasalle Road, Suite 636
Towson, MD 21286

July 9, 2012

CONTENTS

APPENDICES

1.0 SYSTEM ARCHITECTURE

1.1 Features Overview

The ERTAC EGU growth committee requested that the software development be done using languages readily available on common Linux distributions and that the system code be open source and available to the states. Since the EGU hourly datasets contain many gigabytes of data, there was a need identified to use database management software to effectively handle the large amount of information needed for growth projections. An additional need for this system was the capability to create reports and data exports in an easily-read format such as CSV (comma-separated values) that can be used by spreadsheet programs such as Microsoft Excel and by other databases such as Microsoft Access. Finally, it was expected that many different people in separate organizations will want to install and use this system, so the necessary software should be readily available, and not too complex to setup and operate.

1.2 Software

SQLite database software was chosen to manage the data, along with the Python programming language to perform the various processing steps. These choices were influenced by the following factors. SQLite was selected for the database component rather than MySQL or PostgreSQL primarily because SQLite is simpler to install and administer. The system does not require the capabilities of a multi-user client-server database, so there is no need to burden the users with that kind of complexity. The database is created in the computer memory, used to process data and export the output files, then removed from memory. The approach of having the database loaded into memory rather than stored on the hard drive reduces the input/output load and provides increased processing speed.

The Python language was chosen because it is a widely-used programming language which will allow ERTAC to modify their growth projection system in the future, and the Python database API supports database bindings for all of MySQL, PostgreSQL, and SQLite, as well as other database management software. Both SQLite and Python also include good support for reading and writing data in CSV format to allow transfer to other systems.

This system was developed to run under Linux, using Python and SQLite. However, users quickly began testing in a Windows environment. Because of this, the CSV data loading routine was updated to be more tolerant of file format differences between Linux and Windows. Python and SQLite are also available to install and run under Windows, so the same model source code can be run without modification on Windows just as on Linux.

The main software components are:

- A recent version of Linux from either the developer community or a commercial enterprise.
- A recent version of Python from the 2.6.* or 2.7.* series. Python 3 includes some incompatible language changes which this model does not support.
- A recent version of SQLite from the 3.6.* or 3.7.* series should be included with the Python libraries. It is not necessary to install a separate command-line version of SQLite, although doing so should not cause any harm, either.

Because of execution errors reported by some users under older versions of Python, the program debugging information was changed to display the version of Python, the SQLite3 module for Python, and the SQLite3 database library version.

For people using older Linux versions (such as Red Hat Enterprise Linux 5.*) that have system dependencies on older versions of Python which cannot be upgraded, it is possible to install Python 2.6 or 2.7 in addition to the older version, without replacing the system default. It is also possible to install a newer version of Python just for an individual user, without affecting other users of the system.

1.3 Hardware

It is expected that there is a broad range of computer hardware currently being used to run other models by the same users who will want to run ERTAC's growth projection system. Here are some guidelines about how different hardware options are likely to affect performance:

- All else being equal, newer faster CPUs will improve model speed (naturally), but hyperthreading, multi-core CPUs, and multiple-CPU systems are unlikely to provide much improvement since the ERTAC algorithm is inherently not parallel. Decisions about load levels for a particular unit must be made before any units that fall later in the assignment hierarchy, so the entire model is made up of several sequential iterative phases. Still, having other cores or CPUs available to handle other tasks would still be somewhat beneficial, since other processes will be running on the same computers as the ERTAC model.

- There may be some slight benefit from running in 64-bit mode instead of 32-bit mode, because 64-bit addressing does allow use of more than 4GB of RAM. Having a lot of RAM available for data caching is generally helpful for performance, provided that it is properly configured. We have seen systems that performed poorly because they had slow mismatched memory modules installed, and when matched sets of faster modules were installed in a fully-interleaved mode the measured RAM bandwidth improved by nearly three-fold.

- The amount of disk storage needed will depend on how many different scenarios any particular person or agency wants to keep available, since each dataset will occupy many gigabytes of disk space. Higher-speed drives, and multiple drives configured in some kind of striping arrangement (such as RAID 0, RAID 5, and RAID 1+0) will improve the I/O performance while reading these large datasets. In addition, if the input and output data storage is done on separate spindles than /tmp that will reduce I/O contention during some of the processing phases.

2.0 PROCESSING REVIEW

2.1 Pre-Processing

Before beginning a model run, the user must prepare several different input files for data import. Some of these files may be developed by the user in Excel or some other software program, and must be converted into “comma separated value” (.csv) text files for input to the pre-processing module. Other files, such as the CAMD hourly data, are already in CSV format and do not need any modification. The input files for the pre-processor are listed in section 3.1, while the input files for the main projection module are listed in section 5.0. The table and field descriptions for all of the external data files used by the model are included in Appendix D.

NOTE: If using Excel to edit CSV input files for the EGU model, you must take care to avoid incorrectly transforming ID fields (which aren't used for arithmetic) into numeric fields. If you double-click a CSV file and allow Excel to open the file using its default behavior, any string of digits which "looks like a number" will be converted into a numeric format. This conversion will unfortunately remove leading zeros from ID fields such as ORIS plant ID, unit ID, FIPS code, ZIP code, etc. which will corrupt the data. To avoid this undesirable behaviour, you should run Excel and then use its file-opening dialog to open the CSV file. This will allow you to specify that the ID columns should be treated as text rather than as numbers.

A simple command line interface is used to create a run specification identifying the file names and locations for input and output data files. Control is passed to the pre-processing module, which performs edit checks and preliminary calculations of the input files and produces a log file showing data validation issues that must be corrected by the user prior to continuing the run. If data corrections are necessary, they should be made in the original input files and the pre-processor should be run again until any issues with erroneous input data are resolved. In addition to the data edit checks, the pre-processing module performs calculations to fill in derived data fields in the input files and generates an updated set of intermediate data files in the necessary format to pass on to the main projection phase of the model.

NOTE: The output files from pre-processing provide important information to assist the user in identifying needed input data corrections or adjustments. These data corrections/adjustments are necessary to ensure that the projection model output is valid.

Once all edit check issues are resolved, the pre-processing module generates files with the necessary input data and parameters used to execute the main processing module. These files include updated copies of the original input data, additional derived tables for the temporal and unit hierarchies for load assignment, proxy loads to be applied to new units, and generation parameters for the projected future year.

2.2 Main Processing

The main projection processing module is the "core" portion of the model and receives the principal input from the pre-processed files. The projection module performs the growth calculations, determines if any generic new units need to be created because of insufficient capacity, and distributes any excess generation pool to available units subject to hourly and annual operating limitations. The projection modules also calculates future emissions from each unit, including the effects of any specified emission control devices that will be installed. The output from the projection module includes a run log file to identify any data problems that were identified during this stage, and several tabular report files in CSV format listing details and summaries of the projection results.

3.0 USER INSTRUCTIONS

Running the ERTAC EGU software is a two step process that requires the user to execute a separate command line Python script for each step.

1. The first script to be executed is “ertac_preprocess.py.” This is the preprocessor that reads and validates raw comma separated value (CSV) formatted input data files.

2. The second script to be executed is “ertac_projection.py.” The projection processor reads the input files generated by the preprocessor and models the projection scenario.

The preprocessor program reads the input files into a SQLite database, calculates several updated data tables needed for the next processing phase, and writes out the intermediate data in CSV format for review, along with the preprocessor log file in text format.

The projection program reads as its input the intermediate CSV files that were created by the preprocessor. That data is placed into a SQLite database, and future operating levels and emissions are calculated for each hour of the projected future year. The results of that phase are written out into the final data files in CSV format, together with a projection log file.

Of the source code files that make up the model, only the ertac_preprocess.py and ertac_projection.py files need to be made executable with chmod +x. The other source code files are not directly executed, but are used by the two main programs for the first and second phases. The preprocessor and projection model source code may reside anywhere on the host system, and do not have to be in the same directory as the input data.

The input data files for a single run should be placed in a single directory, although it is acceptable to have symlinks in the input directory that point to actual data files in other

locations. This may be useful in cases where multiple projection scenarios are going to be tried for the same base-year data, so that multiple copies of the large CAMD hourly data files are not needed.

3.1 Input/Output Filename Conventions

Input data files should be exactly named:

camd_hourly_base.csv
ertac_control_emissions.csv
ertac_growth_rates.csv
ertac_hourly_noncamd.csv
ertac_initial_uaf.csv
ertac_input_variables.csv
state_total_listing.csv
group_total_listing.csv

The names of the input and output files are hard coded filenames that are meaningful to the purpose of each. However, to preserve uniqueness and distinguish between input and output files for multiple modeling scenarios, a runtime option can be used on the command line during program execution to enable the user to specify different filename prefixes for either input files created by the user and/or output files generated during execution. This protects files from being overwritten from one model run to the next, if the user wishes to preserve the files from a previous run. This also allows the user to uniquely name each set of input and output files with a name that is meaningful for the projection scenario modeled.

To use this approach, the script name is followed by a *–i* switch and the desired filename prefix for input files, and/or a *–o* switch and the filename prefix for the output files. See section 3.2 for example syntax.

Another approach to keep separate run results is to put the output files in different directories and use the full path to the input and/or output files when running the command line program execution.

3.2 Example

In this example, both the preprocessor and projection scripts are being run for a projection scenario in which no prefix is being used for the preprocessor inputs, the filename prefix for all output generated by the preprocessor (which will also be input to the projection script) is "Projection1_prep_", and the prefix for all output files generated by the projection script is "Projection1_model_". This example assumes that the ERTAC projection model source code is located at: *{path_to_code}.*

Set the working directory to the directory where the input files are located. At the command prompt type:

cd {data_directory}

Run the preprocessor. At the command prompt type:

*{path_to_code}/ertac_preprocess.py -o Projection1_prep_*

After reviewing the preprocessor run log, named Projection1_prep_ertac_egu_log.txt, for any problems, run the projection script. At the command prompt type:

*{path_to_code}/ertac_preprocess.py -i Projection1_prep_ -o Projection1_model_*

The input prefix (-i) for the projection phase must match the output prefix (-o) that was used in the earlier preprocessor phase, or else the projection will not be able to read the right set of files and will fail.

Note: For a list of runtime options that may be specified on the command line when the preprocessor or projection script is executed and the usage of each, at the

command prompt type the filename of the script followed by a space and *-h* switch. For example, type:
{path_to_code}/ertac_preprocess.py -h

The switch options fall into three groups:

a. Listing the available switches:
-h or --help

b. Changing the extent of on-screen messages:
-d or --debug
-q or --quiet
-v or --verbose (recommended default)

c. Specifying prefixes for input and output filenames:
-i or --input-prefix=
-o or --output-prefix=

We recommend using the verbose on-screen messages (separate from the run log file) to monitor which processing steps have been completed. For large model runs which might take several hours to finish, it may be helpful to run the program within a "screen" session, so that you can disconnect while the model continues to run, and reconnect later to resume monitoring the status messages.

Note: For running the model under Windows instead of Linux, the source code directory and data directory can be prepared in almost exactly the same way. The model code will successfully read input data files with either Linux/Unix line endings (LF) or DOS/Windows line endings (CR+LF). Under Windows, the directory separator character is a backslash (\) instead of a forward slash (/), and the path to a directory might also need a drive letter specified, if the source files

and data files are stored on different drives. So, to run the preprocessor example that is listed above on a Windows computer, at the command prompt type:

*{path_to_code}\ertac_preprocess.py -o Projection1_prep_*

.

4.0 PREPROCESSOR VALIDATIONS

The preprocessor initially loads input data from external CSV files into SQLite tables, performing basic format checks on numeric and date values, and verifying that required fields are present. The data type information in ertac_tables.py is used to perform these checks. Integer and floating-point values are checked for valid numeric entries, and date values are checked against several possible valid formats, including fully-qualified dates, year-only, and day-month. For the online and offline dates in the UAF, if only a year is specified instead of a fully-qualified date, the date is treated as January 1 of that year. For the base year and future year, only a 4-digit year should be specified; likewise for the year applicable for the state and group emission caps.

After the input data has been loaded, a number of additional cross-table consistency checks are then run:

- Check that the base year and future year specified in ERTAC_INPUT_VARIABLES and in ERTAC_GROWTH_RATES are consistent, and that all the hourly data reported in CAMD_HOURLY_BASE and in ERTAC_HOURLY_NONCAMD comes from the base year. Any difference here indicates that the input files are mismatched and must be corrected. This is considered to be an unrecoverable fatal error, and the preprocessor will halt.
- Check that the regions and fuels to be processed are consistent among ERTAC_GROWTH_RATES, ERTAC_INITIAL_UAF, and ERTAC_INPUT_VARIABLES. Mismatches found here may not be fatal, but could skew the future generation and emission estimates. Only the sets of regions and fuels that exist in all three of these tables can be projected into the future year. For example, if a region-fuel combination is present in the UAF and in the input variables, but not in the growth rates, it would be impossible to project that region-fuel's future generation demand.
- Check that facilities are consistent among CAMD_HOURLY_BASE, ERTAC_HOURLY_NONCAMD, ERTAC_INITIAL_UAF, and ERTAC_INPUT_VARIABLES. These tests identify facilities in the input variables (for placement of generic units) that are not in the UAF, and facilities in the hourly

data that are not in the UAF. If there are facilities listed in the input variables that do not exist in the UAF, then it would be impossible to look up the plant data needed for creation of generic units. If there are facilities included in the hourly data that are missing from the UAF, then the generation and emissions data for all units at those facilities will be excluded from the model.

- Check that facilities and units are consistent among ERTAC_CONTROL_EMISSIONS, ERTAC_INITIAL_UAF, CAMD_HOURLY_BASE, and ERTAC_HOURLY_NONCAMD. These tests identify units in the control/emission table that do not exist in the UAF, and units in the hourly data that do not exist in the UAF. If there are units listed in the control/emissions file that are missing from the UAF, no future emissions can be estimated for those units. If there are units with hourly data but no entry in the UAF, then generation and emissions data for those particular units will be excluded from the model.

After the cross-table checks, another set of intra-table consistency checks are run to validate multiple-field interactions within the data tables.

- Check UAF for consistent data
  - Facility-specific data should be consistent across multiple units at the same facility
  - Unit online dates should precede offline dates
  - If there are multiple records for units that switched fuels, they should not have overlapping date ranges
  - Units marked as NEW should come online after base year and not have a blank online date
  - Units marked as Full or Partial units should already be online before or during the base year
  - Presence or absence of annual capacity limit should be consistent with the capacity-limited flag

- Check growth rates for consistent transition hours
  - Hierarchical hour for changing from peak growth plateau to transitional formula should comes before the hour for changing from transitional formula to non-peak plateau for remainder of year

- Check input variables for consistent new unit sizes
  - New unit minimum size should be smaller than new unit maximum size

- Check control/emissions for consistent dates, and presence of emission rate and/or control efficiency
  - Factor start date should come before factor end date
  - If multiple factors exist for the same pollutant at the same unit, they should not have overlapping date ranges
  - Factor start date should come after base year, otherwise factor will be ignored
  - Either emission rate or control efficiency should be present; if both are supplied, emission rate will be used later in projection phase for emission calculations
- Check group total listing for consistent lists of valid states
  - The same group name should have same set of states, if listed multiple times
  - All states should be valid

After the within-table checks, the values in the input data are checked against upper and lower warning ranges (for numeric data), or lists of valid entries (for string data), according to the allowed values specified in the ertac_tables.py file. Any excessively high or low values, or unknown states or fuels, will result in warning messages about unusual input data which might indicate typographical errors or outlier values.

After the data range validations, the preprocessor makes additional checks for regions and fuels where lack of units and activity may not allow reliable projection of future generation.

- Check for only NEW units in region/fuel with no base-year activity
- Check for region/fuel with fewer than 10 units active in base year
- Check for region/fuel with all units retired in future

In addition to the earlier data checks, the preprocessor will also produce warning messages during later calculation steps if necessary data is missing or impossible values would result from the computations.

- During temporal hierarchy processing, if the hierarchy code for a region/fuel is missing or unknown.
- During calculation of non-peak growth rates, if the root-finding algorithm cannot find a numerical solution for the non-peak rate value, or if the rate would result in negative generation.
- During heat rate calculation, if an existing unit has no heat input or gload in the base-year hourly data, so the heat rate can't be calculated. Also, if units without a calculated heat rate do not have a specified nominal heat rate, either.
- During maximum heat input calculation, if there was no actual heat input data, and there was no supplied generating capacity to convert into heat input.
- During optimal load calculation, if there was no gload in hourly data, so optimal load can't be calculated.
- During utilization fraction calculation, if there was no heat input in hourly data, so utilization fraction can't be calculated.

5.0 PROJECTION MODULE

After the pre-processor has performed its data validation checks, determined the temporal and unit hierarchies, filled the initial values in calc_generation_parms, calculated the non-peak growth factors, and established the proxy loads for the planned new units, it writes out the intermediate results into a series of CSV files which will be available for input to the projection module. Those files, with optional prefixes specified by the -o switch for the preprocessor, are listed below. The table and field descriptions for these files are included in Appendix D.

- calc_hourly_base.csv
- calc_updated_uaf.csv
- calc_unit_hierarchy.csv
- calc_1hour_hierarchy.csv
- calc_6hour_hierarchy.csv
- calc_24hour_hierarchy.csv
- calc_generation_proxy.csv
- calc_generation_parms.csv
- calc_growth_rates.csv
- calc_input_variables.csv
- calc_control_emissions.csv
- calc_state_total_listing.csv
- calc_group_total_listing.csv

After reviewing the preprocessor run log to confirm that there were no problems that need to be resolved, then the projection module can be run to compute the predicted future generation and emissions from all the EGUs.

The assignment of future generation loads is done in a multi-step procedure, as described in the process diagram in the "ERTAC Implementation #18.ppt" file (reference file Chapter5_ProjectionModule_ERTAC_Implementation.zip).

First, existing units are loaded based on their base-year operating levels and the hour-specific growth factors, subject to hourly and annual operating limits for each unit. The total generation load on the existing units and the proxy load on the new units is evaluated to determine if excess generation needs to be distributed among the units within a region-fuel block, or if there is enough capacity deficit to require the creation of one or more generic new units, in which case the growth step will be restarted.

After the generation growth is finished, any excess generation pool at each operating hour will be distributed across all available units within the region-fuel group, in unit hierarchy order, subject to hourly and annual limits on each unit. The first distribution pass will increase unit loads only up to their optimal hourly level; if excess generation still exists after all units have been raised to optimal loads, then a second pass will raise unit loads up to their hourly maximum values if necessary.

After all the generation assignment has been completed, the available spinning reserve will be evaluated for each region, to determine if sufficient reserve capacity exists within the region across all fuel types.

The projected future emission values are then calculated based on the predicted heat input at each hour for each unit, using base-year emission rates and taking into account any specified changes to rates or control efficiency that would be effective in the future year.

A number of detail and summary reports, as described in Appendix B, Reporting Functions, are produced by the projection module and exported as CSV files at the end of the processing.

6.0 Q & A

This section includes questions and answers and discussions that came up during the development and testing of the EGU system.

Question 1.

How should non-EGU units be handled?

Answer 1.

A "non-EGU" entry should go into the hourly data type field (column S) to indicate that the model should ignore any CAMD hourly data coming from that particular unit.

Putting non-EGU units into the main UAF sheet and flagging them as non-EGU will tell the model to ignore them. For nonexistent or retired units, if there's no CAMD hourly data reported for those IDs then there isn't a need to have any UAF row to indicate their disposition. On the other hand, there would be no harm caused by having UAF rows with pre-2007 offline dates if you did want to keep all the information together for completeness.

Any excess EGUs which have data in CAMD but aren't represented in the UAF will result in warning messages about the mismatches, because the model doesn't know what the user wants to do with the data from those units. The unmatched units aren't used in any calculations; the model doesn't know anything about them (notably region/fuel bin) so their base year activity can't be used anywhere and is discarded. The reason this produces a warning message is that the program doesn't know whether the omission was intended or was a mistake. In either case, the generation and emissions from the unmatched units won't be included in the base year or projected to the future year. Also, having the ORIS/Unit ID's labeled as non-EGUs does provide some clarity in the data set.

The unit hierarchy does not include any non-EGUs.

Question 2.

How should cases be handled where new units are specified in the UAF for region/fuel combinations which had no hourly data in the base year?

Answer 2.

In this situation, there is no temporal profile available for the new units, and there is no base-year generation to which growth can be applied. It was decided that the guidance should be that the modelers should manually move these units to different nearby similar regions, which have existing base-year hourly data for that fuel. This would allow the new units to follow the temporal profile of existing units, although the demand growth would not be based on the correct region.

Question 3.

What data must be entered in the UAF to satisfy the required-data checks for non-EGUs?

Answer 3:

There are 7 entries needed in the UAF to satisfy the required-data checks; everything else could be left empty to save effort on your part. The required columns are: the plant ID and the unit ID, the state and the facility name (which should match what's in the CAMD data for that unit), the ERTAC region and fuel bin (which need to be filled but don't have to be "right"), and the hourly data type ("Non-EGU"). If those columns are all filled then there should not be any spurious warnings if the remainder of the non-EGU rows are left empty.

The MAX_ERTAC_HI_HOURLY_SUMMER is not treated as a required field in the input UAF, so the initial data loading and validation procedures don't complain about missing data here. However, there is another check for MAX_ERTAC_HI_HOURLY_SUMMER in the CALCULATE_HEAT_INPUTS procedure, and that is what is producing the message you're seeing. That extra test was added on April 10 as one of several requested additional data checks in the preprocessor. Apparently nobody has noticed this side-effect of the added test until now.

In this specific case, since the projection module won't use the non-EGU data for any calculations, you could simply ignore this warning for those particular units. If you'd like to suppress the warning message for those rows, you could put a dummy filler value into the MAX_UNIT_HEAT_INPUT column, which will be used as an alternate source for MAX_ERTAC_HI_HOURLY_SUMMER in CALCULATE_HEAT_INPUTS, and that will satisfy the check for missing data at that point.

Question 4:

For a test case of all the coal fired existing units in VA with no retirements and no new units as well as a growth rate of 1, there are hours listed in the base_retired_generation column of calc_generation_parms.xlsx that contain retired generation, from 1 MW up to 735 MW. Why?

Answer 4:

The base_retired_generation column is affected not only by retired units, but also by units with capacity limits specified, which produce non-0 values in that column.

Question 5:

In what kind of situations would units be turned off in an hour of the future year despite having been on in the same hour of the base year.

Answer 5:

First, units sometimes have base year heat input reported without any steam load or gross load; if there is no projected load, the future year heat input drops to 0, reducing the number of operating hours. Second, for any hours where AFYGR has become 0 (typically meaning that proxy loads on new units completely satisfy projected demand) the existing units would be off, again reducing the number of operating hours.

Question 6.

ORIS codes with the ERTAC numbering methodology were added to the rows previously without ORIS codes for the new units. Some ORIS codes have different ERTAC regions, county names and lat/long coordinates. The differing data is from the state spreadsheets.

Answer 6.

Looking at the first few rows mentioned, it looks like plant IDs have been filled in but unit IDs left blank. Both the plant ID (for the entire facility) and the unit ID (for a particular unit within that facility) are needed in order to process the information for any specific unit. For the first rows listed, which are both units at the same plant (which seems to be a new facility) you'd probably want to assign the same new plant ID for both rows, and assign two distinct new unit IDs that are unique within that plant.

Although the projection model is not affected by inconsistent county or lat/long information for a plant, the differences could indicate problems with the data that the states provided. A single facility might cross county boundaries. The lat/long is supposed to be a single set of coordinates for the entire plant, but some states (notably Texas) seem to have supplied unit- or stack-specific coordinates instead, with several different values at the same facility. If they didn't follow your directions for that data, they may also have made mistakes in other values.

The complete set of these inconsistent entries is found in the QA_ertac_egu_log.txt file in the section that starts with "Warning: UAF has inconsistent details for ORIS plants:"

The projection model *is* affected by inconsistent assignment of units at a single plant to different ERTAC regions. This means that the generating capacity of that plant is divided among separate regions. That means that the evaluation of spinning reserve requirements for each of the regions may not be accurate. Also, if a generic new unit needs to be created at such a plant, the assignment of the new unit to the correct ERTAC region could be problematic.

Question 7:

If the state supplies the heat rate, how is it used?

Answer 7:

If there is a state-supplied heat rate in the nominal_heat_rate column, then that overrides the use of the calculated heat rate, and the state's value is used.

Question 8:

What happens when there are ORIS/Unit IDs that are in the CAMD hourly file but not in the UAF?

Answer 8:

For ORIS/Unit IDs that are in the CAMD hourly file but not in the UAF, any ORIS/Unit ID that was in the CAMD hourly file but NOT in the UAF is removed from the CAMD hourly file prior to any preprocessing. The removal of the unmatched hourly data doesn't appear explicitly in the

log file, but the row count when the calculated hourly data is written out will be less than the row count from the initial input if some of the units weren't matched. However, if there are also added non-CAMD rows, deleted non-EGU rows, and synthesized partial-year rows all changing the amount of hourly data, it can be a bit harder to follow the counts.

Question 9.

A heat rate value of 837,000 with a UAF check limit of 20,000 means we have a bad value, or a bad warning limit. The heat rate value of 837,000 was from the state spreadsheet.

Answer 9.

With a warning limit of 20,000 we have either (a) a limit that's set much too low, (b) a unit that's more than 40 times less efficient than anything expected, or (c) wrong data provided. The program has no way to know which; all it does is log the warnings so someone can review the data, and correct it if needed.

Question 10.

Is there more information about the facility/units in the control file that are not in the UAF?
Relevant Error/Warning: “Warning: XX facility/units in control/emissions data did not match any ORISPL_CODE, UNITID in UAF:"

Answer 10.

The unmatched IDs from the control file can be seen in the section that starts with "Warning: 57 facility/units in control/emissions data did not match any ORISPL_CODE, UNITID in UAF:" in the QA_ertac_egu_log.txt file. Those IDs don't look like they would be for any new units, so I'm not sure if this updated UAF will resolve any of these problems. Since those IDs don't exist in

the UAF, any CAMD data for those units would not be included in the base year, and the control information for those units would not be applied in the future year either.

Question 11.

I changed all but coal's transition points to 10/50, however, that also produces a warning since those values are outside of the expected range.

Answer 11.

The lower and upper limits for each of the transition hours can be changed in the ertac_tables.py file to eliminate these unwanted messages. In that file now the first transition (peak-to-formula) is expected to be somewhere between 0 and 500 hours (in demand order, not chronological order) and the second transition (formula-to-nonpeak) is expected between 50 and 4000 hours; there is also a separate check that the first transition is earlier than the second. You can also set the transitions independently for each region/fuel combination. For example, one region could have their Oil transition hours set at 40/200 while another region uses 100/600 for the same fuel, if the growth characteristics need to have different time profiles.

Question 12.

"Unmatched CAMD" contains CAMD units not in the UAF. This is ok since some of them are shutdown, new, non-EGU, etc.

Answer 12.

I have to disagree about this particular case. If there was hourly data in 2007 from a unit that was shut down later, there should be a UAF record to indicate when that unit went offline; there shouldn't be any 2007 data from a post-2007 new unit; any non-EGU data in CAMD should have UAF records marking the units to be ignored.

Question 13.

There are two warnings repeated frequently for some WV units for which I either need some clarification or for which there might need to be some adjustments made.

Answer 13.

The old warning levels had seemed to be much too high for some fields, and too low for others, when we talked about some of them in September 2011. There was a suggestion about finding the maximum reported values from the 2007 and 2010 CAMD data, and setting a warning level at about 95% of the max. So, the GLOAD upper level was reduced from 15,000,000 MW (roughly comparable to the total amount of electrical generation world-wide) to 1,300 MW; on the other hand, the SO2_MASS upper level was increased from its original value of 50,000 pounds to 81,000 pounds. Any of those limits can be changed in the code.

Question 14.

Validating regions and fuel bins for input variables, growth rates, and UAF. Warning: regions and fuel bins found in input variables, but not in growth rates:

('CA-N', 'Oil')
('CA-S', 'Oil')
('ERCT', 'Oil')

In the growth rates file, all three regions and fuel unit types appear to be listed. They all have annual growth rates but not peak growth rates. Is the error message looking for the data to be filled in rather than for the existence of the region and bin?

Answer 14.

The reason that the preprocessor is complaining about ('CA-N', 'Oil') and the others not being found in the growth rate table is that the data for that particular region/fuel combination wasn't loaded from the growth rate CSV file into the growth rate database table. That happened because the peak growth rate, which is a required field, had no value specified in the CSV file for some rows. That was noted a little earlier in the log file, at the line starting with "File: ertac_growth_rates.csv line: 8 -- Can't use bad input row; missing data in one or more required columns: ['Peak Growth Rate']" Since those incomplete rows were not put into the table, the later cross-table consistency checks complain about mismatched region/fuel combinations that exist in some tables and are missing in others.

Any time that the initial portion of the run log shows that an input data file was missing a required field, or had invalid data where a number or date was expected, that means there is a problem which needs to be corrected in the input files, and then the program should be re-run. Ideally, the only messages you would see in the top of the log file when the program is loading input data would be about the recognition of header rows, and the number of rows read in from the various CSV files into the tables. Any data errors at this stage can skew the results of all the later steps.

For these three particular growth rate rows, the data fix that is needed would be to fill in some value for the peak growth rate. It does not matter to the program whether the peak growth value is higher or lower than the average growth (since both possibilities can occur), only that the value is present in the input data file. I think some of these missing values were discussed in August, but a decision about what should actually be filled in may never have been made.

Also discussed in that same time-frame was the issue of whether the transition hour values (typically 200/2000) for the growth rate function should be changed, particularly to have earlier transitions for some of the non-coal fuel types. Was there a final decision on that issue? If you're going to be updating the growth rate input file to fill in the missing peak rates, it seems like that would be a reasonable time to also update the transition hours in the same file.

Question 15:

The case is 8 coal fired existing units and 1 coal fired new unit that is small (about 50 MW). I artificially set the max ertac HI hourly summer for the first unit in the unit hierarchy (3797, Unit 5) about equivalent to the maximum hour recorded in the base year data so that there would definitely be generation assigned to the excess generation pool in at least a few hours.

The preprocessor gave the following error:
Calculating non-peak growth factors.
Warning: Secant root-finding failed: impending division by zero for ('VAPW', 'Coal')
Transition hours to formula and to non-peak rate may need to be set earlier.

I set the peak and the annual growth rate to 1.04, and the program seems to calculate the nonpeak growth rate to be 1.04, which is as I would expect. What is causing the problem or what is the program is trying to tell me?

Answer 15:

The root-finding algorithm that the preprocessor uses to compute the unknown non-peak growth rate needs to have two different initial estimates, so it can calculate how far off the projected growth would be using each estimate, and then compute an improved estimate of the non-peak growth rate. It repeats this process iteratively until a satisfactory NPGR is found, or some other condition causes the algorithm to exit early.

We use the specified values of the peak growth rate and the average growth rate as the first two estimates of the non-peak growth rate to initialize this process. In the case you described, both of those values were identical, so their functional values would also be identical. This leads to a condition where delta X = 0 and delta Y = 0 and the slope of the line between the two points is undefined because the points coincide. That is what the warning message is telling you.

Since you apparently wanted to have the peak, average, and non-peak growth factors all be the same in this case, the last estimated value (1.04) is usable, but the warning could cause unnecessary concern. If there are likely to be real cases where the projected growth is supposed to be identical for all hours, we could add a test in the code to look at whether the first two estimates are identical, and if so change one of them by a small factor, say 1.000001 for example. The root finder would then converge to the desired value after the first step without encountering an undefined 0/0 value.

Question 16:

How can base year data be added, modified, or deleted by using the

ERTAC_HOURLY_NONCAMD file?

Answer 16:

Doris: By including a non-CAMD file, you can:

*Overwrite various hours in the CAMD database for existing units, an hour at a time, or
*Add hours for units that maybe don't report to CAMD but the hourly data exists for some other reason.

So, in the case of AR, there were 9 units that did not have any gross load, or did not have any heat input, or did not have both gross load or heat input in the BY. One was retired, it seems, so I pulled that one out. For the other 8, I added one line of data for the first hour of the year in the nonCAMD file (see attached). I used just a very little amount of power (10 MW) for each, and a heat input that was based off a heat rate of 10,000 btu/kw. If you didn't want to guess at all this, you could go into the 2010 or 2011 CAMD data, and pull actual data for one hour of operation out of those files to use in this file.

The program wrote over that one hour of data in the BY with the lien item in this file, providing the unit with one hour where all the data was available for calculations. I made sure in the UAF

that each unit had a reasonable nominal_heat_rate and a reasonable max_unit_heat_input so that the extra hour of data wouldn't change any of those values.

Seemed to work just fine. These units were available in the future year to pick up whatever excess generation was available. I've attached the unit_level_activity.csv file. You can see that each of these units have really small utilization fractions in the base year.

I think this might be the best way we've come up with to handle these units that were idle in 2007, but are still going to be viable elements in the electrical grid in the future, at least until we can update the code to better handle these type of units.

Robert: The optional ERTAC_HOURLY_NONCAMD data file has the same format as CAMD_HOURLY_BASE, and when those files are processed the data rows from the non-CAMD file will be added to the CAMD data, or will replace rows where the 4-part key (plant, unit, date, hour) from the non-CAMD file matches the key in the CAMD file. Merging the hourly data in this way allows any or all of the following tasks to be accomplished:

1. Hourly data for an EGU which was not included in the CAMD file can be added, by filling in the entire year's data for that unit in the non-CAMD file. If the unit is properly entered into the UAF then it will be processed just as any other unit.

2. Partial-year reporters which do not have a complete year of data in the CAMD file can have data for some or all of the non-reported months put into the non-CAMD file. This allows more accurate analysis of the non-reported part of the year for those units, instead of assuming a constant flat load for all of the non-reported hours.

3. Individual missing hourly measurements can be filled in adding relevant rows to the non-CAMD file. Occasionally the CAMD file will have an hourly row missing from a full-year unit, and the non-CAMD file can be used to fill in such gaps.

4. Erroneous hourly CAMD data can be corrected or erased, by creating a non-CAMD record for the same (plant, unit, date, hour) as the invalid data. If correct values are known, they can be filled in the non-CAMD row and will replace the incorrect CAMD data. If valid measurements can't be determined, all the non-key columns of the non-CAMD row can be left empty, which will effectively erase the incorrect values in the CAMD file.

All of these kinds of hourly data modifications can be done by putting the specified entries in the non-CAMD file, which should typically be much smaller and easier to edit than the CAMD hourly data.

Question 17:

How should the nominal_heat_rate column be used versus the ERTAC_heat_rate column?

Answer 17:

*Unit Heat Rates: By now we all know that making sure each unit has a valid heat rate is pretty important. However, I've found it is much better to type in the unit specific, state supplied heat rate (for new or existing units) in the **nominal_heat_rate** column, rather than the ERTAC_heat_rate column. If a value is typed into the ERTAC_heat_rate column, it will supersede all other data. Also, when you look at that input file at a later date, it is more difficult to tell if that ERTAC_heat_rate data is code-calculated or user-supplied. I put all my user supplied heat rates for units in the **nominal_heat_rate** column, and deleted out all the data in both the ertac_heat_rate column and the calculated_BY_average_heat_rate column. This makes for a much more readable UAF, and allows the code to do all the other filling in. The nominal_heat_rate will still take precedent over anything else, but it is a much cleaner way to input the data and be able to understand what the code is doing.

Question 18:

How should the max_unit_heat_input column be used versus the max_ertac_heat_input_hourly_summer column?

Answer 18:

*Unit Maximum Heat Inputs: Again, we all know by now that each unit has got to have a max_ERTAC_heat_input_hourly_summer. However, it is much better to type the unit specific, state supplied heat input (for new and existing units) in the **max_unit_heat_input** column, rather than the max_ertac_heat_input_hourly_summer column. Typing the data directly into the max_ertac_heat_input_hourly_summer column seems to override any other comparison the code might make. By typing the data in the max_unit_heat_input column, the code will still compare the max_unit_heat_input column with the hourly_base_max_actual_heat_input to come up with the max_ertac_heat_input_hourly_summer. If there is no base year data, the program would use the max_unit_heat_input data. And again, by deleting out all the apocryphal data from hourly_base_max_actual_heat_input and max_ertac_heat_input_hourly_summer, the UAF is a lot easier to figure out. This comparison between actual base year data and the state supplied max may not be terribly important now because we've run 2007 so many times, we are pretty familiar with it. But eventually, when switching base years, having that data in the max_unit_heat_input column will allow the program to possibly readjust the max_ERTAC_heat_input_hourly_summer based on the new base year data.

Question 19:

What kind of data cleanup is needed, e.g., for EGU versus non-EGU line items?

Answer 19:

The warning messages noted in Julie's June 8 email identify a couple of facilities where UAF rows for different units at the same facility have slightly inconsistent facility-level descriptive

data. In this case the difference is just the presence or absence of the state/county FIPS code, which should have no impact on how the model runs.

The edit checks will warn about any detected data issues that might potentially cause problems, but the program can't exercise human judgement to decide whether any particular issue is actually serious enough to need corrections. That's why we rely on people to read the log messages and evaluate the nature of the warnings. In this example, a person can readily see that the differences for separate records from the Newark Bay Cogen facility are minor and can be ignored; likewise for the Eagle Point facility. You don't have to keep changing the input data until there are absolutely no warning messages in the log files; you just need to make sure none of the warnings are important to you.

In your June 11 email you mentioned that some of the units at these two facilities were flagged as non-EGU in the UAF. That doesn't have any impact on the consistency checks. When you actually deleted one of the three rows for 50385 from the UAF, you apparently removed the single inconsistent row and left the two matching rows in place, thus eliminating a warning for that particular facility ID. If you look closely at all the UAF rows for 55113, I think you'll find that some of them have the FIPS codes filled in and some don't, which is what causes the warning about inconsistency for that facility, independently of the non-EGU status of those units.

In Doris's June 11 email she mentioned that you had also encountered warnings related to inconsistent plant data in the CAMD hourly files. As above, if the differences are minor you can decide to just ignore the warning and proceed. That's probably a safer choice than manually editing the very large base-year hourly data input.

I wouldn't worry about minor changes in descriptions, such as whether a plant name sometimes has "Inc." at the end and sometimes does not. I would worry about whether operating parameters are significantly outside of normal ranges, such as heat rates of 1500 or 150000. The software isn't smart enough to decide how important any possible problem might be; all it can do is point them out so you can determine which issues need attention.

Question 20:

What should be done with units that had no activity in the base year that were not retiring?

Answer 20:

Doris: I think the semi-consensus between Robert and I is that right now, the code is not flexible enough in the input files to allow a really quick and elegant way of keeping these units in the file. The biggest problems are that without base year data, the code doesn't have a way of assigning a base year utilization fraction or a unit optimal load threshold. Neither of those columns has a state override mechanism, and both are calculated off the base year data. So, with no base year data, a division by zero error occurs. Robert, jump in if I'm incorrect or describing these items wrong.

Unfortunately, we in the implementation committee just did not foresee this data issue. Should have, but didn't.

I think the way to include those is to put maybe 10 hours of "fake" data using the nonCAMD input file. Because I'm forgetful, I'd just make it the first 10 hours of the year, and set every hour to the same value, maybe 50% of the capacity of the unit. If it would help, you could send me an ORIS/Unit ID of a problematic unit, and I could gin up an example nonCAMD input file for you. Would that help?

Julie recently asked me for my top 5 listing of items I'd like to see fixed, if we ever get any money for another contract. This issue was in my top 5, for sure. While it didn't happen in VA data for 2007, I'm sure eventually it MIGHT happen in any particular state or region. I'd hate to have to fuss around with fake data when some state overrides on those columns should allow the program to utilize the unit for power distribution in the future year.

Robert: There is another approach you could try, instead of creating fake hourly data for these existing but non-operating units. In some circumstances, you would be able to fill in the desired operating values for these units in the initial UAF that is input to the preprocessor, so that the projection phase of the model could assign some generation to them when distributing the excess generation pool. Naturally, with no base-year activity reported for these units, no generation would be assigned to them during the growth steps on page 2 of the block diagram, but they would still be eligible for the distribution steps on page 3.

There are four operating parameters (max_ertac_hi_hourly_summer, max_annual_ertac_uf, ertac_heat_rate, unit_max_optimal_load_threshold) needed for the projection code to allocate generation to these units. Some of these have explicit state override columns defined in the UAF table, while for others you can fill in the desired value in the column where the preprocessor would have put its calculated result if it could.

1. For max_ertac_hi_hourly_summer, if there is no actual heat input data in the base year (i.e. the CAMD hourly data only has blank or 0 values for the unit for the entire year), but there is a value supplied in the UAF column for max_unit_heat_input, then that value will be copied into max_ertac_hi_hourly_summer by the preprocessor. If neither a calculated nor a supplied value is available to use for max_ertac_hi_hourly_summer, a warning message will be printed.

2. For max_annual_ertac_uf, if there is no actual heat input data in the base year, but there is a value supplied in the UAF column for max_annual_state_uf, then that value will be copied into max_annual_ertac_uf by the preprocessor. There will be a warning message about the fact that the utilization fraction could not be calculated due to the missing heat input data, but that does not prevent the use of a supplied utilization fraction instead.

3. For ertac_heat_rate, if there is no gload and/or heat input data in the base year, but there is a value in the UAF column for nominal_heat_rate, then that value will be copied into ertac_heat_rate by the preprocessor. There will be a warning message about the fact that a

heat rate could not be calculated for the unit, but that does not prevent the use of a supplied heat rate instead.

4. As you indicated, there is no state override for unit_max_optimal_load_threshold. However, when there is no gload data in the base year and the optimal load cannot be calculated, any pre-filled value in the UAF column for unit_max_optimal_load_threshold will be left unchanged. There will be a warning message about the fact that the optimal load could not be calculated, but the supplied value will be passed on from preprocessor to projection. So, even though there is no state override column in the UAF for optimal load, you can still put in a value that will be used.

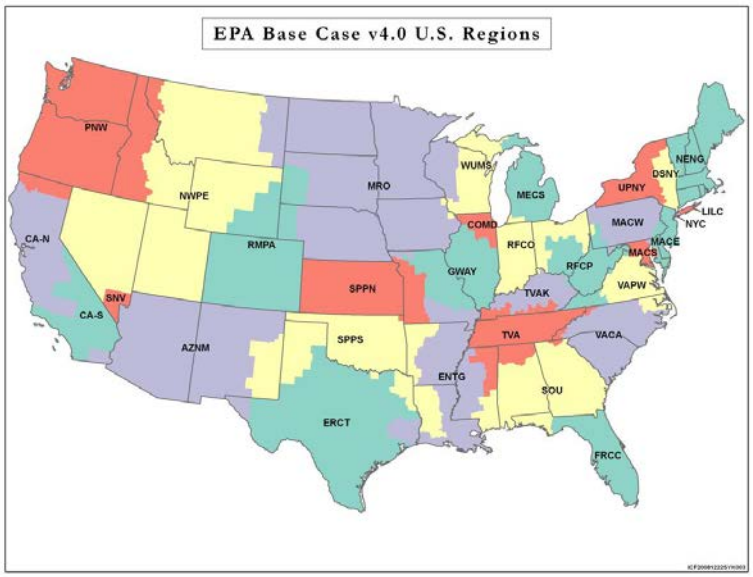
You may want to compare the kinds of results you get from using fake CAMD hourly data versus filling in these UAF override values. One difference that occurs to me is that by putting in only a few hours of fake data (to avoid skewing the base year) you would end up with very low UF values for these units. That would result in most of their future capacity being used in the growth steps for the hours with the fake data, leaving those units with little capability to help with the excess generation pool, which is where I think you actually want them available.

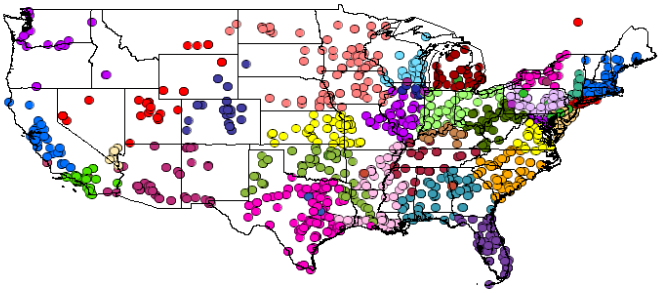
Question 21:

How can we divide model runs covering a wide region into a few smaller regions so that we are not overwhelmed by extremely large and unmanageable outputs?

Answer 21:

Two plots were done showing demarcations of ERTAC regions. The ERTAC EGU code is run by these regions along with fuel/unit type bins. The first map, “EPA Base Case v4.0 U.S. Regions is an original document (by EPA?). The second map is a plot that was made by using the latest version of master UAF maintained by Wendy. All units in Wendy’s master file are plotted and grouped in different colors by what’s listed in “ertac_region” column.





Comparing the two plots, it's easy to see that state boundaries have become unimportant. An ERTAC region usually spans multiple states. For example, RFCP (dark green in 2nd attachment) covers OH, WV, VA and MD. Conversely, a state could belong to two, three or more ERTAC regions. Texas is an example of this, having spanned SPPS (green), ERCT (magenta), and ENTG (pink). (The two blue points are probably data error).

Since the ERTAC code currently does not allow generation transfer across regions, model runs should be conducted by regions, not by states. However, there will be some minor difficulties, because CAMD data is organized by states (not by ERTAC regions). In addition, CAMD data is quite large and the code is designed to read whatever CAMD data one supplies. CAMD inputs containing a bunch of unnecessary units not only slows down model runs, but also generates many warning messages in the run log.

Here is what we could do. Extract from Wendy's master file ERTAC regions of intended model coverage. Find out how many states there are in the extracted file. Pull CAMD data only for those states. Ideally, some scripts should be developed to pull CAMD data for only the absolute necessary units in the UAF, and nothing else. This will speed up model runs and make log file cleaner.

Question 22:

For combined cycle gas plants, which outputs file should be examined to investigate why generic units are being created?

Answer 22:

There are a number of items and files to examine to figure out why generic units are being added. It requires a lot of digging. When the capacity in UAF based on the sum of the max_ertac_heat_input_hourly_summer columns is not enough to satisfy the demand for that hour, a new unit is added. The demand for any hour is based on the sum of the gross load for each unit, converted to heat input using the heat rate.

*Look at the calc generation parms file to figure out what are the highest hours in the hierarchy. If the generation needed in that file appears to be higher than what you believe the universe of units is capable of generating, then the unit should be creating new units. The states at that point may want to go in and add a state supplied unit that satisfies the expected need.

*Look at the UAF file and check out the unit heat rates. As noted in the first paragraph, the heat rate is used to convert between the heat input and the gross load. If the ERTAC heat rate is real high, say 15,000 – 20,000 btu/kw-hr, then what happens is that the gross load, when converted to heat input, becomes inflated. This may drive up the hourly demand for heat input to the point where the sum of the needed heat input looks to be greater than the available heat input capacity. If this happens, a new unit is generated. So, to fix this problem, the state can go into the UAF and supply reasonable heat rates in the nominal_heat_rate column. This will override the calculated heat rates.

Question 23:

What guidance was the MANE-VU states given about updating their control file data?
Answer 23:

The Eastern Regional Technical Advisory Committee (ERTAC) is in the process of testing the EGU growth projection model code, and in order to support the testing effort, MARAMA has another data review request for the MANE-VU region. About a year ago you were asked to review control file spreadsheets. The control file spreadsheets are used to assess unit emission rates and/or control equipment control efficiency. Significant changes have been announced for many facilities in the last year. Therefore, the ERTAC workgroup is asking that you review the attached control file spreadsheet for your state and update data in the following columns, based on what you currently know about your CAMD facilities. For units that are not expected to have significant changes from the base year (2007) into the future, no updates need to be made.

In this spreadsheet, please consider these ideas when making the updates:

- For existing units, if no emission rates or control efficiencies are included in the controls file, the program will use the base year emissions data to estimate future year emissions.

- For new units, if no emission rates are provided in the controls file, the program will try to estimate what the new unit's emission rates might be from base year data for existing units. Therefore, if future year emission rates can be estimated for new units, please try to include those rates if at all possible in the controls file.
- The following list provides the minimum needed data for the program to include emission rate data in its calculations. Please note that for existing units, if only one pollutant has significant emission rate changes, data only needs to be provided for that pollutant. For example, if a unit put on an SCR, but not FGD, between the base year and the future year, only columns A, B, O, P, and either Q or R need to be filled in.

For units with base year (2007) data:

Column A - oris plant id

Column B - camd unit id

Column H - SO2 Control Start Date

Column I – SO2 Control End Date (if there is an end date – otherwise this is not mandatory)

Either Column J - SO2 Control Efficiency (from base year actual emissions data) or Column K - Controlled SO2 Rate (lbs/mmbtu)

Column O - NOx Control Start Date

Column P – NOx Control End Date (if there is an end date – otherwise this is not mandatory)

Either Column Q - NOx Control Efficiency (from base year actual emissions data) or Column R - Controlled NOx Rate (lbs/mmbtu)

For new units (started operating after 2007):

Column A – oris plant id

Column B – camd unit id

Column H – SO2 Control Start Date

Column I – SO2 Control End Date (if there is an end date – otherwise this is not mandatory)

Column K – Controlled SO2 Rate (lbs/mmbtu)

Column O – NOx Control Start Date

Column P – NOx Control End Date (if there is an end date – otherwise this is not mandatory)

Column R – Controlled NOx Rate (lbs/mmbtu)

Question 24:

How can pollutants other than NOx and SO2 be used in the program?

Answer 24:

This version of the processor only processes NOx and SO2. The hourly_diagnostic_file format only has columns where emissions for NOx and SO2 can be stored, so the projection code only

calculates values for those pollutants. You could conceivably add a post-processing step to be run after the projection code, where another program could read in the hourly_diagnostic_file that had been created, calculate emissions for additional pollutants, and write out a wider version of the hourly_diagnostic_file that was augmented with extra columns for all of the other pollutants. I think it would be much more reliable if you created a program to do that, as opposed to manually pasting in the formulas for those calculations with Excel.

Question 25:

The preprocessor is giving me the following for some very old coal units in Ohio that are not being run much:

Warning: UAF has inconsistent capacity limit values and flags:
('2837', '1', 'Coal', , 'Y')

I have about 20 of them in OH and wanted your advice on what route to take with them, should I pull them or have the state send me more data? What does this flag really mean to the model?

I noticed that the preprocessor only throws the flag if there is no max_annual_ertac_uf_State Input variable and a "Y" under capacity limit. If both variables are present then it does not. So I should just remove all the capacity limited flags and max_annual_ertac_uf_State Input for all those that have it and save for another day? My other option is to ask the state for the uf data for those units that are causing the flag. I think this is something we would also want to discuss when we get ready to do the East of the Mississippi run down the road. It is not many units in the Midwest.

Answer 25:

I think what that message means is that there is a Y in the capacity limited field in the UAF, but no state supplied utilization fraction in that column for that unit in the UAF. As background, the units that are marked capacity limited and provided a state supplied utilization fraction were supposed to represent units that perhaps had some sort of federally enforceable mechanism (or other type of mechanism) in the future year that does not allow the unit to be utilized to the default utilization fraction in the future year. As an example, VA had a facility that received a federal cap on emissions in a federal consent agreement and that cap was significantly below the amount of emissions that the facility's three units emitted in the base year. The facility reps told me that while they were switching to a low sulfur coal, they were also planning on just not running the units as much in the future to meet the cap, and that would have meant that they units could run at about 70% utilization in the future, rather than the 90% that was the default. So, to tell the program this, I put a Y in the capacity limited column and a 0.7 in the state supplied utilization fraction column.

HOWEVER, now that I've told you all this, Jin and I have discovered that this functionality doesn't work the way it was intended. My fault, really, because the documentation (and who was the dummy that wrote the documentation? Oh, yeah, that was me….) didn't clearly define how the program was supposed to do this and Robert coded it incorrectly from the get go. It would have been a major task to recode it, so the group decided to let it go. It is on the list, however, to be fixed.

So, due to the fact that the capacity limitation is not really working, the best thing to do is just get rid of the Y in capacity limitation and if there is a utilization fraction in the state supplied utilization fraction, you may want to get rid of that as well.

Yes, I would definitely suggest just removing both the capacity limited flags and the data under max_annual_ertac_uf_state_input. That function does not work very well at all. It is something that is on the to-fix listing.

Question 26:

What happens when a unit is retired in the middle of a model year?

Answer 26:

Jin and I ran a case called 7-retire. It consists of 9 units (8 existing units and 1 new, state supplied unit) on coal in VAPW, with growth rates of about 1.04. I set it up to retire one unit, which had a UF of about 0.5 in the base year. As you can see from the plots of this unit (ORIS 3797, unit ID #3), the program does seem to handle the unit as we had envisioned. The unit was available for power generation up to its maximum hourly capacity until the retirement date of 5/17/2017. After that date, it was provided zero power. Also, after that date, the retired generation from the unit was included in each of the adjusted future year hourly growth rates, so that the other units in the run make up the difference in generation from the retired unit. It looks like it is functioning well.

Question 27:

Can outliers from data points with less than an hour op_time significantly affect outputs?

Answer 27:

I've looked at the graph and spreadsheet that Byeong sent, and I'm not sure there's actually any problem here.

It appears that these test results were from a model run with a growth factor of 1 and a future heat rate of 9479. This means the FY GLOAD should exactly match the BY GLOAD, and the FY heat input could vary above or below the BY heat input depending on how close the BY heat rate was to 9479 at each particular hour.

In the results, 4 of the fractional-hour data points have low BY heat rates, ranging from 1270 to 6236, so the FY heat input for those hours is higher than the BY heat input, as expected. The same is true for many of the full-hour data points; some of those have BY heat rates as low as 7120, so their FY heat input is also higher than their BY heat input to meet the same GLOAD.

At the other end of the spectrum, there are also several data points with high BY heat rate values (the top 8 ranging from 22697 to 97443), so for all of those the FY heat input is set lower than the BY heat input.

There are also about 30 data points with non-zero heat input but 0 GLOAD in the BY data, some with fractional-hour operation and some with full-hour operation. For all of these, the projected FY GLOAD of 0 sets their FY heat input down to 0. These are the points that you see clumped along the X-axis of Byeong's graph.

As far as I can tell, all of the values in each of these cases have been calculated correctly based on the supplied input data. If Byeong or anyone else computes a different result using the same formulas as the model, I would need to know about that.

I don't know if the CAMD hourly data is any more likely to have unusual values reported at hours with fractional OP_TIME than for the whole-hour data points. There hasn't been much fractional-hour data in the files that I've seen, and I haven't done any specific analysis of those data values, so I don't have any conclusion about whether or not the fractional-hour data is less reliable in general, or if you might want to edit or exclude it.

The model doesn't make any special adjustments for partial-hour BY data. In particular, it does not project partial-hour FY operation; if a unit is operating at all in an hour of the FY, it's presumed to be available for use during that entire hour, so the FY OP_TIME will be set to 1.

Question 28:

In a few test cases that we have run in house so far, I noticed that, if no generic units are created, annual growth rate (specified in the input variable) can be roughly estimated (and thus verified) by dividing annual summation of hourly FY Gload over all units (in a region/fuel unit type bin) by annual summation of hourly BY Gload over the same units. However, the same method cannot be used (i.e., the math does not add up) for cases when generic units are created, even if proxy loads are excluded in the calculations. Do you know why that is? I would think this simple math should hold up regardless of whether or not new units are created.

Answer 28:

In the calc_generation_parms file, the sum of the future_projected_generation/sum of the base_actual_generation always equals the annual growth rate for the region and fuel-unit type bin as specified in the growth_rates file.

In this same file, (the sum of total_proxy_generation + the sum of adjusted_projected_generation)/sum of base_actual_generation should also always equal the annual growth rate for the region and fuel-unit type bin as specified in the growth_rates file. However, sometimes it doesn't. The reason is that sometimes proxy power is assigned in magnitudes that are actually greater than the amount of power projected for that hour. The program does not back down the proxy power being assigned to equal to demand, in these cases. (a big oversight in programming) So, sometimes the calculated annual growth rate for these types of ERTAC fuel unit type bins and regions will actually be larger than expected.

Excluding proxy loading doesn't help because the hour specific growth rate is modified by the amount of proxy power assigned to each unit in each hour. If the proxy power overwhelms the expected demand, the program turns off all existing units, but there still may be more proxy power than is needed.

The results will always be conservative though because I can't figure out any scenario where this glitch doesn't overestimate future demand.

Question 29:

How are emission factors for new units calculated?

Answer 29:

The emission factors for the new units are the values from the 90th-percentile "cleanest" existing unit for that particular pollutant in the base year. There's no interpolation between values, and the individual hourly measurements are rolled up into annual or seasonal averages before this determination is made.

So, in your spreadsheet, the new unit SO2 rate comes from plant 3803 unit 1, whose overall average SO2 rate (column U) of 0.93178 was the cleanest of the 8 units in the base year. Similarly, the new unit NOx rate (all year) comes from plant 3803 unit 3, whose average OS NOx rate (column W) of 0.02782 was the cleanest of the 8 units.

Question 30:

Do units with the highest utilization fraction reach their maximum hourly rates most often?

Answer 30:

Here's an interesting result from the last run. I've always expected the units with the highest utilization fraction (an annual calculation), and therefore highest in the unit hierarchy, to be the units that reach their maximum hourly rates most often.

Turns out, that might not always be true.

I took 8 existing coal units from the VAPW data set, knowing that all of them are base load units that operate a good bit. I used the current growth rates established for coal in VAPW (1.05 peak and 1.02 annual), with the 200 and 2000 transition points. This run didn't recognize either of the new units I had supplied, and so the new units were just ignored. The run was basically just the 8 existing coal units. As I skimmed through the unit_level_activity file, I noticed that one unit, ORIS 3797 Unit 4, had hit its maximum 63 times. No other unit had hit the maximum, and no new units were created.

What was really interesting to me was that 3797 Unit 4 was ranked 7th for the existing units in the unit hierarchy. None of the units in front of it ever got to its maximum.

Basically what happened is that 3797 Unit 4 operated more hours near its maximum in the base year, but overall operated less than the other 6 units on an annual basis. This just seemed counter intuitive to me, but I checked the math as well as the base year files and the diagnostic files, and that does seem to be what happens.

Here are some of the number I looked over:

<u>3797 Unit 4:</u>

Unit had a supplied heat rate of 9500 btu/kw-hr (about what was calculated by the program). Maximum heat input (max_ertac_heat_input_hourly_summer) was 1,761 mmbtu/hr, which makes the maximum generation in the future year for any hour 185.4 MW (1761 *1000000/9500/1000). On peak hours, where the growth rate is 1.05, the unit could only operate to about 176.5 MW in the BY, otherwise, 185.4 would be exceeded in the FY. It does appear from the base year hourly files that this unit had a significant number of hours of operation in the 177-180 MW range.

3797 Unit 5:

This unit was first in the hierarchical order. It had a supplied heat rate of 9300 btu/kw-hr, again, very close to what the program would calculate as an average from the base year data. The maximum heat input was 3604 mmbtu, or 387.5 MW. At peak hours the unit could only operate at about 369 MW in the base year or the maximum in the future year would be triggered. The base year data shows the unit never operated in the BY at more than 356 MW. But, it operated a whole lot of hours in the range of 330-356, making its UF a good deal higher than Unit 4's.

3803 Unit 4:

The same pattern is true for this unit, which was second in the hierarchical order. It had a supplied heat rate of 9700 btu/kw-hr, which is nearly what the program calculated. The maximum heat input was 2482 mmbtu/hr, or about 256 MW. At a growth rate of 1.05, the unit's BY data needed to be at 244 MW to exceed this level. The base year data shows a significant amount of time operating in the 220-230 range, but the maximum GL never exceeded 230.

These interactions between heat rate, gross load, heat input, FY and BY data, and growth rates can get me confused until I put it all on paper, with the FY and BY data scrolling in front of me on the computer screens. However, I think the point here is that just because a unit is "base load" doesn't necessarily mean that all of its operations will be very similar to other "base load" units. Each unit might have its own quirks, either from data anomalies or from the way the unit was actually operating.

Question 31:

A State questioner asked why are we adding variability limits to the state budgets? If all the state budgets are being fully incorporated into the modeling where are the extra allowances coming from to increase each of the State budgets for variability?

Answer 31:

The state and group tables can contain any set of data the user of the model prefers. What is actually in the tables does not affect how the code will distribute generation at all. Suppose, for instance, that PA had just really tiny assurance levels, well below what is being predicted for the future. The model will not adjust or in any way reduce the FY generation/activity assigned to PA units based on that assurance level. FY generation/activity at any facility is a function of (a) BY activity (b) AEO and NERC growth rates for that particular region (c) the amount of generation retired by the user of the model (d) the amount of new generation added by the user of the model and (e) the amount of new generation added by the model itself to meet future year demand gaps. FY emissions are based on the estimated FY heat input and either 1-the emission rate supplied for that unit in the controls file or 2-the program's estimate of what the FY emission rate will be, based on the BY emission data.

What the model will do is calculate the estimated future year emissions and provide a comparison by state or group to any level that is provided in the state or group charts.

What EPA-CAMD (Kevin Culligen, who is no longer there, so boy is this information dated) has told us in the past is that for the results of the model to be considered a viable/SIP approvable solution, the results of the model need to demonstrate that the tenants of CSAPR are met. Each state can't exceed its assurance levels, and each group of states can't exceed the budget (although some allowance needs to be made for banking of emissions, but that is another story).

So, suppose the model provides data showing that PA can't meet its very tiny assurance levels. The user would have to go back to PA and ask for more control/fuel switch information until the results do show compliance with the assurance levels. This is a clumsy approach; originally the goal was to have the program assign "generic" controls if such an event occurred. However, we ran out of money. It's on the list for upgrades, though.

So, what I would suggest is to include the assurance levels for each state in the state chart and the regional budget levels in the group chart. For the solution to be viable for use in any SIP, both summations must show compliance, I think, or pretty near.

Question 32:

Is the control file similar to the UAF in terms of columns needed to get through the pre-processor and processor? If so, do we know which columns are necessary? Then we could prepare an e-mail similar to the one that we sent to MANE-VU state contacts on the UAF last month.

Answer 32:

Attached is an example.

The required columns I think are:

Orispl_code
Unitid
factor_start
Pollutant_code

For existing units with base year emissions data, the state can fill in EITHER emission_rate OR control_efficiency. If both is filled in, emission_rate will be used. Control_efficiency is the control efficiency from the base year actual emissions data. For instance, if in 2007 the unit was actually emitting at 0.3 lbs NOx/mmbtu, and SNCR is expected by 2018 that will generally achieve 0.2 lbs/mmbtu, the 2018 control_efficency is 33%.

If a state puts data for new units into the file, which is highly encouraged, the state must fill in emission_rate. Control efficiency won't do any good since there is no data for new units to apply the control efficiency.

So, for the State Data tab, which is what the states saw last time and is a much prettier format, the states need to update the following columns:

ORISplant id
Camd unit id
SO2 control start date
So2 control efficiency or controlled SO2 rate (lbs/mmbtu)
NOx control start date
NOx control efficiency or controlled NOx rate (lbs/mmbtu)

If a unit is a new unit, then the state must give the controlled SO2 rate and the controlled NOx rate in lbs/mmbtu

End dates need to be filled in if there is actually an end date. Otherwise, those are not mandatory.

All the other columns are informational and possibly helpful, but not necessary for making the program run.

Question 33:

Does the program perform this way: Unit A has reached its annual limit "yet" in the sense of processing order, because we've already assigned all of its possible generation for the entire year on page 2, and aren't going to shuffle that around any further. That means we're not going to shift any of unit A's generation from hour 5000 to hour 10, so we don't need to change the cumulative values that were already established at all the intervening hours.

Answer 33:

Yes, in the distribution of the excess generation pool the projection code now looks at each unit's end-of-year status to see if they reached their max cumulative heat input by the last hierarchy hour, or if they still have some available headroom. This means that fewer units may be available to accept additional load at earlier hours, if they were already given a full annual load during an earlier pass.

Question 34:

What happens if a unit has neither GLOAD nor SLOAD reported for the entire base year?

Answer 34:

If a unit has neither GLOAD nor SLOAD reported for the entire base year, then the calc_BY_average_heat_rate can't be calculated. If no nominal heat rate was supplied for such a unit, then the ertac_heat_rate is undefined, and it's impossible to correctly assign future load or heat input to that unit.

A warning message was added to the preprocessor to identify such cases, similar to the warning that was added earlier about units with undefined utilization fractions due to no heat input reported in the CAMD data. That way you can either specify a nominal heat rate in the UAF, or else remove the unit from the model altogether.

Question 35:

What are the effects of new units on regions with small populations?

Answer 35:

Jin ran the combined cycle VAPW setup for me, with the "real" growth factors of 1.02 annual and 1.3 peak. I set the peak hours at 30 and 100. We are calling it Case 5-CC-2.

This UAF has 24 units, 19 of which operated in the base year. 1 unit is a nonEGU, which gets stripped out of the calcs. 1 of the existing units (one of the smaller units) is retired in 2015 (not really, but I wanted to make sure all was working correctly in regards to retirements.) 5 units are new units coming on line by 2017 (these are real units, with permits, construction going on, etc). The 5 new units account for a lot of generation capacity, about 1900 MWs or so. Proxy power for these 5 new units was based on the unit in the hierarchy with the second highest UF of the existing units, 3804 6A. The UF for 6A in the BY was 0.3871.

So, this is a "real" region/fuel unit type bin and "real" growth factors, where the only change I made to the data was to retire one small unit. I took a swag at the peak and annual transition hours (as a side note, we do need some guidance, I think, about how to figure out where to set those numbers.)

Anyway, here's what happened:

In some hours of the year, the proxy power supplied to the 5 new units is actually more than the entire estimated future year power needs for that particular hour. All existing units are turned off in the future year when this happens; the new units remain at their proxy loading.

Because the 5 new units are operated at their proxy capacity in these hours, these hours actually have MORE generation assigned than they need. Annually, the total sum of the future year generation is no longer equal to the base year generation multiplied by the annual growth rate. In the example here, the BY generation annually was 9,462,996.88 MW-hrs. Applying the annual growth rate of 1.02, and the FY generation was estimated to be 9,652,256.82 MW-hrs. The total annual FY generation in the hourly_diagnostic_file is 10,839,193.11 MW-hrs, more

like an annual growth rate of 1.145. The calc_generation_parms file shows that of this 10,839,193.11 MW-hrs, 7,731,379.6 MW-hrs is proxy loading, while 3,107,813.51 MW-hrs is adjusted projected generation from existing units.

An interesting side effect of all this is that even though there is extra generation in the FY, emission rates are very low. I included the permitted limitations in the controls file for the new units, so the majority of the generation is coming from the ultra clean units (most combined cycles are clean to begin with). The FY emission rates are about 27 tpy of SO2 and about 1,018 tpy of NOx. The BY emission rates are about (not exactly, I used CAMD summary reports, not the hourly files, to get these numbers) 1,700 tpy of NOx and about 40 tpy of SO2.

So the downside here is that the model is predicting more generation than we want it to because of the amount of power being assigned as proxy to the new units is sometimes in excess of power requirements in the FY hour. The upside is that even with the excessive amount of power being generated, the new units emit so little that there is still a big reduction in FY emissions.

Question 36:

When are generic units created?

Answer 36:

The program is supposed to look at the first 400 hours (in the hierarchy, not temporally) within a region, and examine the total amount of generation needed for all units in the region for each of those hours versus the total amount of generation available. For the worst case hour in that set of 400 hours (which really should be in the first 24 hours or so, depending on how the hierarchy is created), it should figure out how much generation capacity is needed, if any. Suppose the worst case FY hour needs 4000 MW-hr, but the entire capacity for that ERTAC region and fuel/unit type bin, as listed in the UAF, is 3500 MW-hrs. So, the program needs to create 500

MW-hrs, plus whatever the demand cushion is in the input variables. If the demand cushion is 1.1 or 10%, then 500*1.1 or 550 MW of capacity should be created, using the min and max sizes as listed in the input_variables file.

I think the program is supposed to report every hour noted where there is not enough generation capacity available. The excerpt below seems to indicate that there were 3 hours in which not enough capacity was available, the highest being hour 5270, where a lack of 416 was noted. That hour should have been the "trigger" hour, which caused the creation of the new units. The demand cushion should have been applied, etc.

One could argue that this approach to figuring out generation needed is way too simplistic since if there were really only 3 bad hours in a year, the grid operators would probably figure out a more cost effective way of solving the issue (importing power, demand response, or something else) than building a new plant. However, it should satisfy the CAMD folks' concerns that they voiced early on, and the power guys (Dave Smith/AEP and John Shimshock/GenON) on the implementation committee seemed to think it was not hugely unreasonable.

When we start doing multi state/multi region runs, and we see an inordinate amount of generation being created for these small number of hours, then we might have to refine that logic somewhat. I am guessing that the larger the area encompassed by the region, and the larger the number of units in the fuel/unit type bin, the less this will be an issue, because the chances are greater that somewhere in the system a few peaker units were at lower loads in the base year, and can consume the additional generation in the future year. This is one of the reasons that I am concerned by NY and LI. They have lots of these peaker units, but the units are fairly small (and old/dirty), and they are in a relatively tiny area. When one is operating, most of them are going, and at pretty high rates, I think. It is monstrously difficult to build new units in that area since that set of states seem to be strongly opposed to ever redesignating themselves as attainment/maintenance for any NAAQS (a sentiment I do not understand at all…) regardless of air quality.

Question 37:

I decided to do a quick check of all the facilities that had been marked like this message:

Filling in remaining hours for partial-year reporters:
Unit: ('10866', '13', 'Boiler Gas') has 2160 unrecorded hours, 2160 of those seemingly active.
UAF Annual_HI_Partials: None , Reported heat total: 426014.608
All unrecorded hours will be filled with NULL values.
Filled NULL for 2160 rows.
Unit: ('50627', '20B1', 'Combined Cycle Gas') has 2160 unrecorded hours, 2160 of those seemingly active.
UAF Annual_HI_Partials: None , Reported heat total: None
All unrecorded hours will be filled with NULL values.
Filled NULL for 2160 rows.

They are all in Illinois for some reason so I am venturing to guess that if I remove Illinois data the problem hopefully will be closer to a solution.

Answer 37:

Units with these warnings in the log file are all partial year reporters that don't send reports to CAMD for every month or quarter. Usually they are NOx budget trading program units that have to report 5 or 6 months of data, but don't fall under the ARP for 2007, and therefore don't have to report all months of data. The program identifies these units from the UAF file (the column entitled BY_CAMD_hourly_data_type), with 4 choices: full, partial, nonEGU, nonCAMD). It looks to see if an annual heat input has been provided in the UAF (the column called BY_Annual_HI_for_Partials). If an annual heat input has been provided, the CAMD heat input is subtracted from this annual value, and the remainder is divided up among the non-reported hours. If no annual heat input has been provided, the program assumes zero heat input for the unreported hours.

If the base year moves to 2010, then many of the partial year reporters become full year reporters, because in 2010 many of the partial year reporters became subject to CAIR, and were required to become full year reporters. Yet another benefit of moving the base year from 2007 to 2010,

Question 38:

What's the difference between AFYGR and hourly specific growth rate? I was browsing through some old ERTAC e-mails, and in one of the e-mails, you mentioned these two terms. I could not wrap up my brain quick enough to know the differences between the two.

Answer 38:

This is really good question! The program uses and calculates a number of growth rates, so I'll try to provide a run down here, in the order that the program sees or calculates them.

Annual growth rate: this is an input from the user, and usually it will be the Energy Information Administration number for annual growth for that region and fuel. It is inputted in the growth rates table

Peak growth rate: this is an input from the user, and usually it will be the National Energy Regulatory Commission's number for peak hour growth for that region and fuel. It is also inputted in the growth rates table.

Non peak growth rate: this is a calculated number, based on the annual and peak growth rates, and the hours set for the transition points. To summarize, the annual growth rate sets the future year amount of generation (base year annual generation * annual growth rate). However, we want the peak growth hours to be a little higher, which means that most other hours will have to be a little lower in order for the total amount of generation in the year to still be equal to the product of the base year annual generation*the annual growth rate. You may remember this graphic, to illustrate the concept. The peak growth rate is 1.07, the annual growth rate is 0.95,

and the nonpeak growth rate is about 0.933. This value is calculated by the preprocessor and is located for any region and fuel/unit type bin in the file called calc_growth_rates

Hour specific growth rate: The hour specific growth rate is the growth applied to any hour, based on the hierarchy of that hour. So, in the above graph, the hour specific growth rate is equivalent to the peak growth rate for the first 200 hours; it's equivalent to the non peak growth rate for hours 2001 to 8760. For the hours between 200 and 2000, it is equivalent to the value along a linear slope between the two. It is assigned by the preprocessor to each hierarchy hour in the file called calc_generation_parms.

Adjusted future year growth rate: This value is the hour specific growth rate, adjusted for the amount of proxy generation applied to any new units, and also adjusted for any retired unit generation for that hour in the base year. The formula is (hourly base year generation * HSGR – total proxy power for that hour)/(base year generation – retired base year generation). The adjusted future year growth can change on an hourly basis since proxy power and retired generation can be different on an hourly basis. This growth rate is the final growth rate that is applied to existing units' base year activity data to estimate future year activity data. This value is calculated and assigned by the processor to each hierarchy hour in the file called calc_generation_parms. Since it is a processor function and not a pre-processor function, the calc_generation_parms file only contains this data after the processor runs through.

Question 39:

After a new unit, generic or state-supplied, gets an initial proxy loading, will the remaining capacity of that unit be eligible for receiving access generation? For example, in the attached spreadsheet you developed a while ago (which, by the way, I consult with frequently. It is my favorite reference to learn ERTAC EGU code algorithm), the proxy for F and G is specified as 75 percent. The remaining 25 percent is free for receiving access generation, correct?

Answer 39:

Yes, that is correct. The remaining capacity of new units not consumed by the proxy loading is available for use in consuming the excess generation pool. In fact, these units should receive a good proportion of the excess generation pool, since the input variables currently require that the new units be inserted into the unit hierarchy at a pretty high level. Therefore, they should receive generation first or nearly first, before older, less efficient units.

Question 40:

Question about calculations: I noticed that the hourly ratio of gross load (MW-hr) to heat input in the hourly_diagnostic_file is not really consistent. Apparently, the ratio should be 10 but I saw some variations. All generic unit seems to show about 10.096 while existing units showed some variability. The new planned unit has 10 exactly.

Answer 40:

The heat input does change from base year to future year, usually. I will try to explain the relationships, but it is actually easier to do with a diagram. I'll attempt to do it with words here. These are actually very good questions; they get to the heart of the entire EGU industry, really.

Heat input (HI) represents the amount of fuel a unit burns. Heat input is a physical measurement of the actual # of BTUs supplied to a boiler or turbine or other device. So, for instance, many units burn coal. Coal has a heat content (a measure of the density of energy within a material) here in Virginia of about 13,000 btu/lb. In other words, when a lb of coal is burned to completion, it will give off 13,000 btu in energy of some type. The goal of an electric generating unit is to turn that latent energy in the fuel, measured as btu, into electrical energy (electrons moving), measured as joules as efficiently as possible. So, heat input measures the amount of fuel going into the boiler, usually in btu; electrical energy coming out of the boiler is measured in joules.

To determine the rate of electricity produced, the # of joules are measured over a time period, joules/second, which are also called watts. For whatever reason, the industry standard measure of electricity is kilowatt-hours, or 1000 Joules/seconds * 3,600 seconds/hr. A rather silly means of expression, but there you go. Mechanical engineering 101. Anyway, the ratio of kilowatt-hrs produced to btus supplied to the EGU is called the "heat rate" and is expressed in terms of btu/kw-hr. This is a terribly important value to everyone in the EGU community since it is essentially an expression of efficiency. Units with high heat rates are shutting down left and right in the new regulatory era. Units with low heat rates (efficient units) are seeing increased utilization. For environmental types, the measure of heat rate in btu/kw-hr is a way of converting from input to output, or output to input. It also is a useful tool for determining the "climate footprint" of a unit, since the more efficiently a device produces electricity, the less carbon in the fuel is burned, creating less CO2. For the ERTAC process, we need the heat rate to convert GW –hrs to millions of BTU (mmbtu) and vice versa. Here's the basic formula : Heat input (btu) divided by heat rate (btu/kw-hr)=generation (kw-hr)

The heat rate is determined by unit in one of two ways. A state can input a unit-specific heat rate in the nominal_heat_rate column in the UAF. If data is supplied in this column, then that data is used in the program for the GW –hrs to BTU conversions. Or, for existing units, if no nominal_heat_rate is supplied, the program will calculate the average btu/kw-hr value from CAMD data in the base year. That goes in the calculated_BY_average_heat_rate column. For the ERTAC_heat_rate, the preprocessor fills in this column using the nominal_heat_rate data if there is any available. If not, it uses the calculated_BY_average_heat_rate.

For any particular unit, the heat rate (btu/KW-hr) value should be constant. However, the heat rate can vary from unit to unit a great deal depending on the unit itself and its operating conditions. An old, lightly used, spreader stoker boiler burning coal like James River might have a heat rate of 15,000 btu/kw-hr. A new, combined cycle natural gas facility, like Warren, may have a heat rate of 6,700 btu/kw-hr. Most bituminous coal fired units here in VA are in the range of 9,000 btu/kw-hr to 12,000 btu/kw-hr.

Since heat input is the measure of the unit's fuel usage, the heat input can change and most likely will change from base year to future year. First, the base year data is adjusted by the future year growth rate. That would change the future year heat input and output in MW. Secondly, the unit in the future year hour may have additional generation assigned to it from the excess generation pool. If additional generation is assigned, then additional heat input must also be assigned.

Question 41:

There is a problem with leading zeros being dropped in Excel that causes a problem when running the preprocessor.

Answer 41:

When you manipulate the columns in Excel files, you should remember that the Unit ID can be alpha-numeric in the CAMD database, so the data in the Unit ID column in the UAF needs to be kept as text. Otherwise, you might lose a leading zero which would cause a mismatch by the preprocessor and an error. For instance, the unit ID may be 092. Keeping the Unit ID column as text will preserve that leading 0. Otherwise, it changes to 92 and the program spits out an error.

Question 42:

What are the steps to create the state UAF?

Answer 42:

To run the preprocessor, you need 4 input files: the UAF, the ertac_input_variables file, the ertac_controls_file, and the ertac_growth_rates file. You also need the base year CAMD hourly data.

When I did the 8-state file set up, I worked one state at a time to try to minimize the errors and warnings. I'd suggest starting with the state for which you have the most background knowledge of the EGU population, which I'm assuming would be Indiana. So, begin by deleting all the line items that are not in IN.

Keep in mind that although there is a boat load of data in the UAF, the preprocessor only needs 7 columns filled in to satisfy the required-data checks. These are as follows: ORIS_Plant_ID, CAMD_Unit/Boiler_ID, State, Facility_Name, BY_CAMD_Hourly_Data_Type, ERTAC_region, and ERTAC_fuel/unit_type_bin. The processor only needs to have a few more columns filled in to actually work. These additional columns are on-line_start_date and Off_line_start_date.

Obviously, ORIS and Unit ID are the keys to identifying units and CAMD hourly data, along with start and end dates. To fill in a variety of the other columns, the preprocessor grabs data from the CAMD hourly file, the growth rates file, and the input variables file, then does some number crunching for max heat input, heat rate, max load, utilization fraction, etc.

Region and fuel/unit type links the unit to particular peak and annual growth rates and also to a variety of input variables.

Facility_Name in the UAF needs to match that information in the CAMD hourly file or a warning flag is thrown. I'm not sure why this is. It could be helpful, I guess.

If a year is entered into the online or off line start date, rather than a month/day/year, the preprocessor automatically changes it to a month/day/year combination.

If you want to add or modify a **new unit**, which is one that was not operating in the base year, you need to add in the 9 columns listed above, and also max_ERTAC_heat_input_hourly_summer (the maximum rated heat input for the unit in mmbtu/hr) and the ERTAC_heat_rate (in btu/kw-hrs). Lastly, there needs to be a Y in New_unit_flag. So, a new unit needs a total of 12 columns filled in. All the other columns are

informational, feed one of these 12 columns, or are calculated values from the pre-processing step.

Next, the UAF may not have all the non-EGU units listed in it. If you want to alleviate as many warnings as possible, then all the units in the CAMD hourly data file need to have a line item in the UAF. If a unit is in the base year CAMD hourly file, but not in the UAF, the program generates a warning in the preprocessor log file. If such a situation exists, the warning will be in the log file, and the preprocessor will strip out those hourly data from the hourly file so that those data are not used in any calculations. Somewhere along the way, the group decided to put non-EGUs in the UAF, and label them as non-EGUs in the column called BY_CAMD_Hourly_Data_Type. This will again strip out the data from the hourly data file, but will provide a cleaner log. If you want to add in a non-EGU to try to minimize warning messages, you only need to fill in the 7 columns noted above. All else can be blank.

One of the number crunching things that the preprocessor does is create an average base year heat rate for a unit that operated in the base year, and it puts the value in the column called ERTAC_heat_rate. It calculates this value from actual generation and heat input data. If for some reason the number is odd, which can happen when the unit is used infrequently, you can override this value by typing a value in the column nominal_heat_rate. It is important to note and correct the odd heat rates in this manner because the heat rate is used by the program to convert GW to Mmbtu and back. A strange number might give all sorts of weird load calculations.

<u>For existing units that operated in the base year:</u>

- The following columns need to be filled in: ORIS_Plant_ID, CAMD_Unit/Boiler_ID, State, Facility_Name, BY_CAMD_Hourly_Data_Type, ERTAC_region, ERTAC_fuel/unit_type_bin, On_line_start_date, and Off_line_start_date.
- The preprocessor uses the base year CAMD data to calculate a unit specific heat rate. If the heat rate is abnormal, which can happen if the unit doesn't operate much or has a

lot of substituted data, then the state can override the calculated heat rate by supply a heat rate in the field called nominal_heat_rate.

- It's ok for the online and offline dates to be years only, if the actual dates are not known. The preprocessor will convert them to month/day/year values.

For new state supplied units:

- The following columns need to be filled in: ORIS_Plant_ID, CAMD_Unit/Boiler_ID, State, Facility_Name, BY_CAMD_Hourly_Data_Type, ERTAC_region, ERTAC_fuel/unit_type_bin. on-line_start_date , Off_line_start_date PLUS max_ERTAC_heat_input_hourly_summer (the maximum rated heat input for the unit in mmbtu/hr) and the ERTAC_heat_rate (in btu/kw-hrs). Also, the New_Unit_Flag column should have a "Y" in it.
- Again, it's ok for the online and offline dates to be years only. The preprocessor will modify them in a conservative fashion.

For nonEGUs in the base year:

- The following columns need to be filled in: ORIS_Plant_ID, CAMD_Unit/Boiler_ID, State, Facility_Name, BY_CAMD_Hourly_Data_Type, ERTAC_region, ERTAC_fuel/unit_type_bin. For these types of units, that is all that is needed to have the preprocessor remove that data from the hourly database and no longer process any of those units.

- The original suggestion to states was to remove the nonEGUs. However, the way AMEC set up the preprocessor, you can minimize errors in the preprocessor log by including the nonEGUs in the UAF and making sure the BY_CAMD_Hourly_Date_Type is nonEGU. I guess we are sort of going in that direction now. By labeling a unit as a nonEGU, it is completely removed from the hourly database and not processed at all, beyond those initial steps in the preprocessor to label and identify them.

The UAF needs to be updated for the most recent shutdowns, fuel switches, and new units. In my opinion, it's probably more important to get the UAF up to date first. If additional controls are being installed, the controls file needs to be updated, but control retrofits just don't seem to be as predominant right now.

I think in regards to capacity limited units, we need to let people know that the capacity limited columns aren't working right now. So, any capacity limitation that is supplied will need to be deleted prior to running the preprocessor/processor. Or, another option is to just remove those columns from the data supplied to the states.

APPENDIX A

NARRATIVE OUTLINE OF DECISION RULES

NARRATIVE OUTLINE OF DECISION RULES

This section is contained in the updated Implementation Outline 10-7-2011.doc, dated 10/7/2011, ERTAC EGU Growth Implementation Subgroup's *Narrative Outline of Decisions and Rules for the Model* (reference file AppendixA_NarrativeOutline_of_DecisionRules.zip).

APPENDIX B

REPORTING FUNCTIONS

REPORTING FUNCTIONS

The projection module produces several reports according to the formats specified in "Reporting Functions draft #4.docx", dated 10/7/2011 (reference file AppendixB_ReportingFunctions.zip). These are all written in CSV format, and include the following reports which are described in that document:

- demand_generation_deficit.csv
- generic_units_created.csv
- reserve_capacity_needed.csv
- unit_level_activity.csv
- cap_analysis.csv
- unit_generic_controls.csv
- capacity_and_fy_demand.csv
- capacity_and_fy_reserve.csv
- state_caps.csv
- group_caps.csv
- hourly_diagnostic_file.csv

In addition to those reports, a group of four tables are also exported at the end of the projection phase, so that the operation of the model and any effects due to creation of generic new units can be reviewed. These files have the same formats as the preprocessor outputs with the same names, as noted in Appendix D. Note that if the preprocessor outputs and projection outputs are not given separate prefixes (using the -o switch with different values) then the projection output for these 4 tables will overwrite the earlier preprocessor output of those same-named tables. These tables are:

- calc_generation_parms.csv
- calc_generation_proxy.csv
- calc_unit_hierarchy.csv
- calc_updated_uaf.csv

APPENDIX C

SOURCE CODE

SOURCE CODE

The separately-included source code for the EGU projection model is made up of five Python files, three SQL files, and two CSV data tables (reference file AppendixC_ertac_egu_20120510_Source.zip).

The two main program files are ertac_preprocess.py and ertac_projection.py, which are directly run for the first and second phases of the model, respectively. The uses of the other files are listed below.

The ertac_lib.py file contains common code which is used during the preprocessing and projection stages. This includes functions for loading and unloading external data files, validation and formatting of data, manipulation of date values, and assignment of proxy loads to new units (both planned and generic).

The ertac_tables.py file defines the expected data types and ranges for input CSV files, and also includes the headings to be used when the data tables are written out from the preprocessor to be passed to the projection module.
The ertac_reports.py file defines the headings to be used in the report output CSV files from the projection module. This is a simpler version of what is done by ertac_tables.py for the input and intermediate data files.

The create_preprocessor_input_tables.sql file is used by the preprocessor to create the SQLite tables that will receive the initial input data.

The create_preprocessor_output_tables.sql file is used by the preprocessor to create the SQLite tables for the calculated and updated data that will be passed on to the projection module. It is also used by the projection module to create its input tables.

The create_projection_output_tables.sql file is used by the projection module to create the SQLite tables to hold the report data that is produced and written out into the final CSV outputs.

The counties.csv and states.csv files are external tables of valid state abbreviations, names, county names, and state and county FIPS codes.

APPENDIX D
DATA DICTIONARY

DATA DICTIONARY

The table and field descriptions for all of the external data files used by the model are included in the separate "Data Files Needed January 6, 2012.xlsx" file (reference file AppendixD_DataDictionary_DataFilesNeeded.zip).

In addition to that file, the following source code files define how the model will read and validate the data files, and the structure of the internal SQLite database tables used by the model.

- The ertac_tables.py file includes the data types and required status for the data columns,and lower and upper limit warning ranges used in validation.

- The create_preprocessor_input_tables.sql, create_preprocessor_output_tables.sql, and create_projection_output_tables.sql files define the SQLite tables used during the preprocessing and projection steps

-

APPENDIX E

SEMAP Presentation

March 14, 2012

This section is contained in “SEMAP Mar 14 2012 ERTAC EGU Growth w/ Notes.pptx” (reference file AppendixE_SEMAP_Presentation.zip).

APPENDIX F

ERTAC Growth Model Improvements

A discussion of EGU Growth Model improvements is included in the document "White Paper topics for EGU growth model.docx" (reference file AppendixF_EGU_Growth_Model_Improvements.zip).

The top three improvements were identified as:

1. Improvement of the unit level activity summary file: For each unit, include also a summary of BY and FY activity (hours operated, HI for OS and Annual, GL for OS and Annual) and BY and FY emissions (NOx OS and Annual, SO2 Annual)

2. Improve proxy generation application: Right now, there are times when proxy generation applied to new units completely overwhelms the region's available generation. Proxy generation to all units in a region should be capped at every hour so that it is never greater than the total amount of generation calculated in the FY for that hour. (This may be ameliorated somewhat when we go to AEO 2012, because the AEO 2012 regions are supposed to generally be larger and have more units, but I still think this will continue to be an issue until it is fixed.)

3a. Handling of units with no base year activity data: At this time, the preprocessor only identifies units with no base year activity data (heat input and/or gross load) so that states can review and remove them manually. However, for units that have no base year activity and will not be retired, it would be better to leave them in the algorithm so that they may be available to meet future year demand. Right now, to do this we have to introduce a minimum amount of "pseudo" data for the base year, which is just a clumsy approach.

3b. Use of optimal minimum thresholds to turn units off when they get too low and redistribute that power: Right now, there is no lower threshold for how lightly a unit may be utilized, which does not necessarily reflect actual operations. An enhancement would be to put lower thresholds on each unit, and turn a unit off if the lower threshold is breached. Power from the unit would then need to be redistributed to other units that have not been turned off.

3c. Application of SO_2 and NO_X controls for caps: Currently the program reports on SO_2 and NO_X emissions as compared to state and regional caps but does not ensure that each cap is met. A useful upgrade would automate the application of controls to meet caps.

APPENDIX G

ERTAC EGU Projection

Model Emails

The email history during development of the EGU Projection Model is contained in reference file AppendixG_EGU_Model_Emails.zip.

APPENDIX H

ACRONYMS AND ABBREVIATIONS

ACRONYMS AND ABBREVIATIONS

Acronym	Description
AEO	Annual Energy Outlook
Btu	British Thermal Unit
CAMD	Clean Air Markets Division (USEPA)
CAP	Criteria Air Pollutant
CEM	Continuous Emission Monitor
CO	Carbon Monoxide
CO_2	Carbon Dioxide
CO_2e	Carbon Dioxide equivalent
EGU	Electric Generating Unit
EIA	Energy Information Administration
ERTAC	Eastern Regional Technical Advisory Committee
FIPS	Federal Information Processing Standard
IPM	Integrated Planning Model
KW	Kilowatt
LADCO	Lake Michigan Air Directors Consortium
MARAMA	Mid-Atlantic Regional Air Management Association
MJO	Multi-Jurisdictional Organization
mmBtu	Million British Thermal Units
MW	Megawatt
NAAQS	National Ambient Air Quality Standards
NEEDS	National Electric Energy Data System
NEMS	National Energy Modeling System
NERC	North American Electric Reliability Corporation
NH_3	Ammonia
NO_x	Oxides of nitrogen
ORL	One-record-per-line (SMOKE Format)
OTAQ	Office of Transportation and Air Quality (USEPA)
OTC	Ozone Transport Commission
PM-CON	Primary PM, Condensable portion only (< 1 micron)
PM-FIL	Primary PM, Filterable portion only
PM-PRI	Primary PM, includes filterables and condensables PM-PRI= PM-FIL + PM-CON
PM10-FIL	Primary PM10, Filterable portion only

ACRONYMS AND ABBREVIATIONS

Acronym	Description
PM10-PRI	Primary PM10, includes filterables and condensables, PM10- PRI = PM0-FIL + PM-CON
PM25-FIL	Primary PM2.5, Filterable portion only
PM25-PRI	Primary PM2.5, includes filterables and condensables PM25-PRI= PM25-FIL + PM-CON
RPO	Regional Planning Organization
SCC	Source Classification Code
SNCR	Selective non-Catalytic Reduction
SCR	Selective Catalytic Reduction
SIP	State Implementation Plan
S/L	State/local
SMOKE	Sparse Matrix Operator Kernel Emissions
SO_2	Sulfur Dioxide
USEPA	U.S Environmental Protection Agency
VOC	Volatile Organic Compounds