# Speciation Tool User's Guide

Version 3.1

Prepared for:
U.S. EPA

Prepared by:
Michele Jimenez
Uarporn Nopmongcol
Greg Yarwood
ENVIRON International Corporation
773 San Marin Drive, Suite 2115
Novato, California, 94945
www.environcorp.com
P-415-899-0700
F-415-899-0707

July 2013

ENVIRON

# ACKNOWLEDGMENTS

The authors gratefully acknowledge Marc Houyoux of the U.S. Environmental Protection Agency's Office of Air Quality Planning and Standards for his contributions to this report.

# CONTENTS

## APPENDICES

## TABLES

## EXHIBITS

# 1.0 BACKGROUND

Air quality models (AQMs) contain a set of equations that use representative lumped modeling compounds to simulate complex atmospheric chemistry. The emission inventory, an important input to the AQM, consists of components such as CO, $NO_x$, total volatile organic compounds (VOCs), and particulate matter (PM). The emission inventory components, also known as pollutants or species, must be converted to the lumped modeling compounds for air quality modeling.

The Speciation Tool is a stand-alone tool that was developed to generate the factors needed to convert the emission inventory species to the AQM lumped species. The conversion process is referred to as *“speciation”*. The factors are referred to as “*split factors”*; i.e., we “split” total VOCs into olefins, aldehydes, etc. The “*chemical mechanism*” is the mapping of the emission inventory pollutants to the AQM compounds.

In the past, most air quality modeling was performed based solely on criteria air pollutants (CAPS). However, more recently the trend is to develop inventories that include both criteria and toxic air pollutants. Toxic air pollutants, also known as hazardous air pollutants (HAPS), are known to cause, or are suspected of causing, serious health effects. The Speciation Tool supports the availability of both CAPS and HAPS. It was designed to integrate the HAPS into the chemical mechanism when that is desired. It is also designed to include both active species and tracer species in modeling; active species are included in the chemical mechanism with chemical feedback, while tracer species are included with no chemical feedback.

The Speciation Tool supports chemical mechanisms for both VOCs and PM.

# 2.0 INTRODUCTION

This user's guide provides instructions for running the Speciation Tool version 3.1. The purpose of the Speciation Tool is to generate the chemical speciation profiles required by emissions processors for converting emission inventory compounds to air quality modeling compounds.

The Speciation Tool supports speciation profile entries for VOC compounds and for particulate matter less than 2.5 microns ($PM_{2.5}$) in diameter based on EPA's SPECIATE 4.3 profile assignments.

- The VOC chemical mechanisms included in the Speciation Tool are Carbon Bond Mechanism versions 4, 5, and 6 (CBIV, CB05, CB6) and the 1999 and 2007 Statewide Air Pollution Research Center mechanism (SAPRC99 and SAPRC2007).
- The $PM_{2.5}$ mechanisms included are the CMAQ version 5 (AE5) and version 6 (AE6) aerosol mechanisms.

The Speciation Tool also produces the factors needed to convert VOC mass to total organic gases (TOG) mass; this is the VOC-to-TOG conversion factor. These factors are needed when an emission inventory is reported as VOC, which is often the case. Because the gas profiles are computed based on TOG while the inventory is reported as VOC, the VOC-to-TOG conversion factor is applied to the inventory VOC to estimate TOG. The speciation profiles are then applied to the TOG value with no loss of mass.

The output files of the Speciation Tool are intended to be used in the Sparse Matrix Operator Kernel Emissions (SMOKE) modeling system. These are the SMOKE speciation profile file (GSPRO) and pollutant-to-pollutant conversion file (GSCNV). The GSCNV file generated by the Speciation Tool is formatted for SMOKE version 2.3 and higher; it will not work for earlier versions of the model. The Speciation Tool does not create the SMOKE speciation cross-reference file (GSREF); you must ensure that the speciation profile codes output by this tool correspond to the codes assigned to emissions sources by the SMOKE GSREF file.

The Speciation Tool is a PostgreSQL database application with a Perl script interface. It is designed for concurrent multiple user access where each run has a unique user-specified name. The Speciation Tool assigns the run name to a database schema for storing temporary tables and results. As a database convention, a schema is essentially a container that stores tables and functions in a selected database. Think of a schema as a folder in the Speciation Tool database. The *shared* schema in the Speciation Tool database, created during initialization, holds the imported data of the mechanism definitions, profile descriptions, profile definitions, and species properties. It also retains the functions that compute the speciation split factors. Before running the Speciation Tool, PostgreSQL, Perl, and Perl libraries must be installed. Speciation Tool scripts must be run to confirm the correct installation of these ancillary tools and to initialize the database. A successful initialization will create the database, load the tool functions, create the *shared* schema with defined tables, and import default data files to the *shared* schema. The instructions for these tasks are provided in Chapter 3.

Chapter 4, "Running the Speciation Tool", provides the details for running the model; run parameters and options are described. A Speciation Tool run creates a named schema in the Speciation Tool database. As illustrated in Exhibit 2-1, initialization creates the *sptoolv3_1* PostgreSQL database with schema *shared,* and each run will create an additional named schema based on the user's specifications. In this example two runs have been made: cb05_criteria and saprc_toxics.

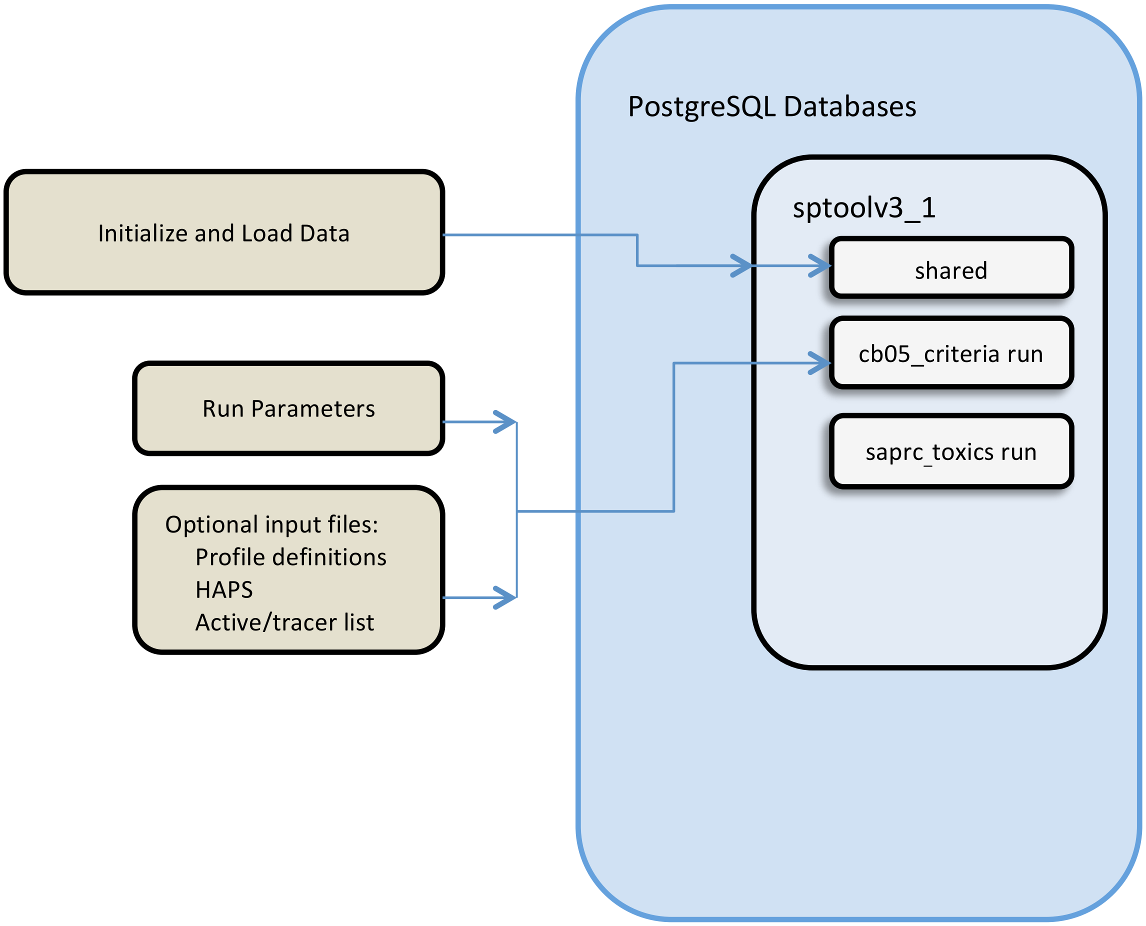

**Exhibit 2-1. Speciation tool initialization and run schemas.**

Chapter 5, "Applications and Methodology", provides additional details on the Speciation Tool calculations. Chapter 6, "Frequently Asked Questions", addresses both basic and advanced questions on using the Speciation Tool.

# 3.0 INITIALIZE THE SPECIATION TOOL DATABASE

The Speciation Tool is written in PostgreSQL using the Perl Database Interface (DBI) and CSV parser (Text::CSV). These software products are open source and available on the internet. You must install PostgreSQL and Perl prior to running the Speciation Tool. Appendix A has instructions on installing these packages and on options required.

The Speciation Tool must be initialized prior to making any runs. Initialization steps include:

- Get the Speciation Tool package
- Extract files
- Set the home directory
- Run the initialization script

Section 3.1 provides a concise step-by-step approach designed for those already familiar with software setup and database applications. Those with less experience can use Section 3.1 as an outline of the steps needed, and then move on to the detailed information given in Sections 3.2 through 3.4.

## 3.1 Quick Start

1. Create a Speciation Tool directory.
2. Extract the Speciation Tool installation files.
    a. Change to the Speciation Tool directory.
    b. Copy the Speciation Tool package to this directory.
    c. Extract and uncompress the zipped file.
3. Verify required software packages are available. See Appendix A for a discussion of the software packages needed.
    a. Run *./sptool_reqd_checks.sh*
4. Set environment variables.
    a. Edit *Assigns.sptool*; set SPTOOL_HOME.
    b. Source *Assigns.sptool.*
5. Initialize the Speciation Tool database.
    a. Execute *./init_sptooldb_v3.1.csh*
6. Execute the provided test case.
    a. Change to the run directory (cd /run).
    b. Run *./cb05_notoxics.job*
7. Review and compare results.
    a. Change to output directory (cd ../output).
    b. Compare results to provided test case files.

## 3.2 Install Speciation Tool

To install the Speciation Tool, create a Speciation Tool home directory from which to work. In the example below it is */sptool,* but you can specify any valid directory name. Copy the installation package file *sptoolv3_1_June2013.tar.gz* to the directory and extract the Speciation Tool files:

```
> tar xvzf sptoolv3_1_June2013.tar.gz
```

After the extraction you should see the files and directories listed in Exhibit 3-1.

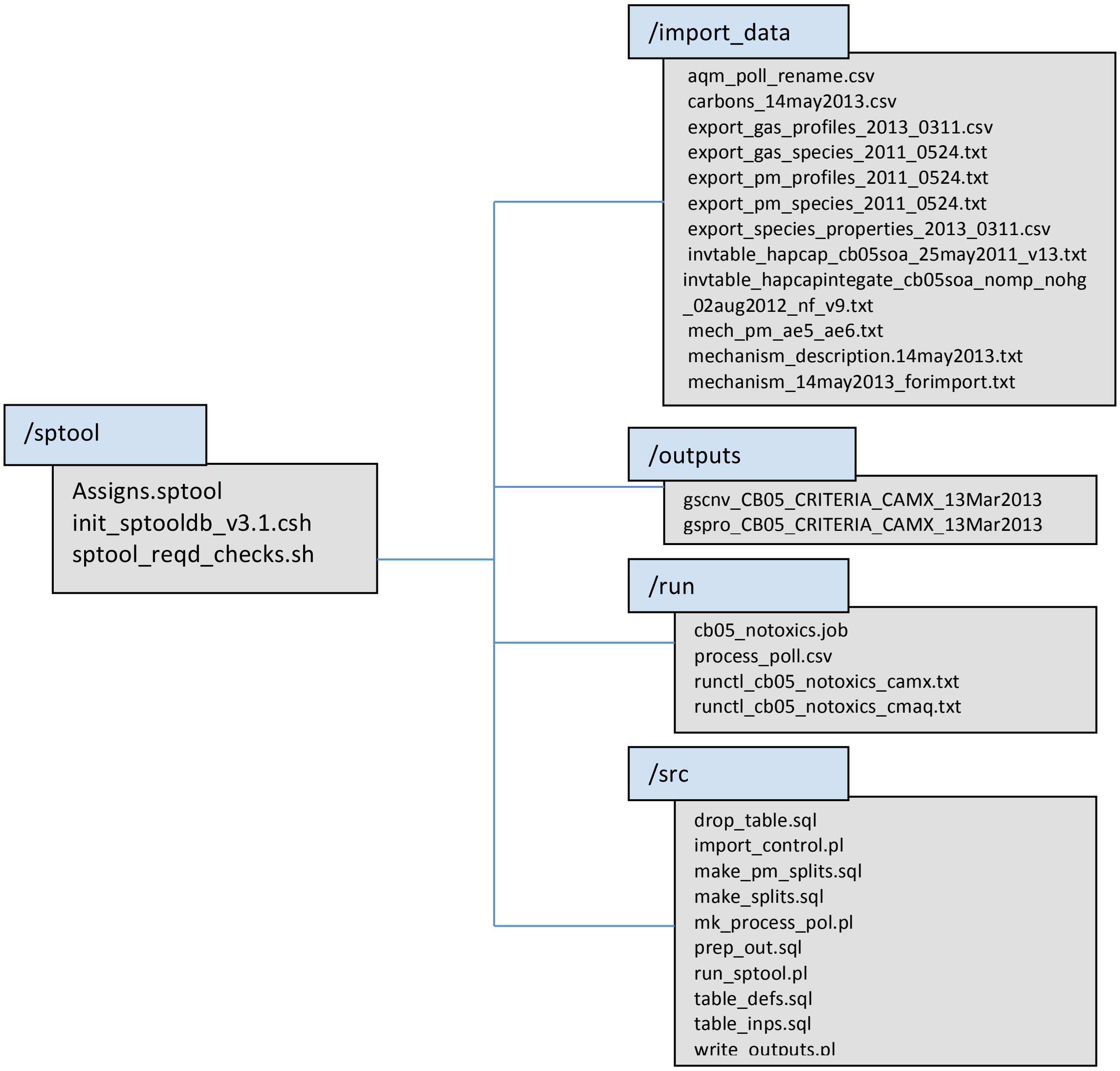
/import_data
aqm_poll_rename.csv
carbons_14may2013.csv
export_gas_profiles_2013_0311.csv
export_gas_species_2011_0524.txt
export_pm_profiles_2011_0524.txt
export_pm_species_2011_0524.txt
export_species_properties_2013_0311.csv
invtable_hapcap_cb05soa_25may2011_v13.txt
invtable_hapcapintegate_cb05soa_nomp_nohg
_02aug2012_nf_v9.txt
mech_pm_ae5_ae6.txt
mechanism_description.14may2013.txt
mechanism_14may2013_forimport.txt
/sptool
Assigns.sptool
init_sptooldb_v3.1.csh
sptool_reqd_checks.sh
/outputs
gscnv_CB05_CRITERIA_CAMX_13Mar2013
gspro_CB05_CRITERIA_CAMX_13Mar2013
/run
cb05_notoxics.job
process_poll.csv
runctl_cb05_notoxics_camx.txt
runctl_cb05_notoxics_cmaq.txt
/src
drop_table.sql
import_control.pl
make_pm_splits.sql
make_splits.sql
mk_process_pol.pl
prep_out.sql
run_sptool.pl
table_defs.sql
table_inps.sql
write_outputs.pl

**Exhibit 3-1. Speciation Tool directory and file names.**

## 3.3 Check for Required Software

Before you begin the initialization, first verify that PostgreSQL and Perl are installed and accessible. Whoever is installing the Speciation Tool must have PostgreSQL permission to create a database. Execute the script *sptool_reqd_checks.sh* to make sure that the required software can be found and to verify user PostgreSQL authorizations.

```
> ./sptool_reqd_checks.sh
```

Following is an example of a successful check for the required software:

```
===== Speciation Tool Requirements Check =====

Checking the status of software requirements...

Status of required software:
[x] = Installed
[ ] = Not installed
[?] = Unable to determine, see notes

------PERL------
[x] Perl
[x]   -DBI
[x]   -DBD-Pg
[x]   -Text-CSV

---POSTGRESQL---
[x] PostgreSQL
[x]   -PL/pgSQL

Refer to the Speciation Tool User Guide Appendix A for installation procedures of the required software.
```

## 3.4 Initialize the Speciation Tool Database

Prior to making any Speciation Tool runs, the tool must be initialized using the steps described below.

### 3.4.1 Update and source the *Assigns.sptool* file

The *Assigns.sptool* file provided in the top-level Speciation Tool home directory sets environment variables that are required for initialization. These include the Speciation Tool home directory, database name, and input file names of the default speciation data. The only environment variable that must be updated is SPTOOL_HOME. This is installation dependent and must be set based on your system. Change the path name to correspond to the directory where the Speciation Tool top directory resides on your system. For example, if you install the Speciation Tool under the directory /disk4/models/emis, you would set the SPTOOL_HOME environment variable like this:

```
setenv SPTOOL_HOME  /disk4/models/emis/sptool  # Speciation Tool top level directory
```

The other environment variable that you might want to change is the Speciation Tool database name. The file provided in the distribution package has this variable set to sptoolv3_1:

```
setenv SPTOOL_DB        sptoolv3_1             # Speciation Tool Database name
```

You may change "sptoolv3_1" to any valid PostgreSQL database name as long as the one you choose does not already exist  The command line 'psql –l' (hyphen lowercase L) will display a list of existing PostgreSQL databases. Restrict the database name's characters to a – z, 0 – 9, and the underscore "_", and use no embedded blank characters.

After the *Assigns.sptool* file has been updated, "source" the file to set the required environment variables.

```
       > source Assigns.sptool
```

### 3.4.2 Create the Speciation Tool database

The initialization script, *init_sptooldb_v3.1.csh*, is a C-shell script that checks for required environment variables, executes a set of PostgreSQL commands to create the Speciation Tool database, sets database permissions, creates the *shared* schema and table definitions for the tool data, and imports Speciation Tool SQL functions and data to the *shared* schema. The functions imported by the script perform the computations needed for generating the speciation profiles and pollutant-to-pollutant factors that are output by the program.

```
        >./init_sptooldb_v3.1.csh
```

Messages are printed to the screen to reflect installation progress. The first part of a successful initialization will look similar to this:

```
>./init_sptooldb_v3.1.csh

SPTOOL_SRC_HOME = /disk4/models/emis/sptool/src
New database: SPTOOL_DB = sptoolv3_1
SPTOOL_USER = yourusername
POSTGRES_BIN = /usr/local/pgsql/bin
CREATE DATABASE
Database sptoolv3_1 created
CREATE SCHEMA
Shared schema created
GRANT
Create permissions granted on sptoolv3_1
GRANT
All permissions granted on shared schema
Defining custom functions and initializing tables.  ...working



 Speciation Tool functions and tables successfully defined in sptoolv3_1.
```

If you forget to set the environment variables, you will get error messages like those displayed below. If this occurs, review the previous section on the *Assigns.sptool* file, source the assigns file, and then rerun the initialization script.

```
SCRIPT ERROR: Required environment variable SPTOOL_SRC_HOME not set
          in script init_sptooldb_v3.1.csh
SCRIPT ERROR: Required environment variable SPTOOL_DB not set
          in script init_sptooldb_v3.1.csh
ABORT: init_sptooldb_v3.1.csh script aborted with errors.
```

If you attempt to initialize the Speciation Tool with a database name that already exists, you will get the following message:

```
createdb: database creation failed: ERROR:  database "sptoolv3_1" already exists
ERROR: failed to create a new database sptoolv3_1. This usually means the database already exists.
          To replace the existing database type 'dropdb sptoolv3_1' from the command line.
```

Delete the existing database if you want it replaced. Otherwise change the SPTOOL_DB name in the *Assigns.sptool* file to a different database name. The command to delete the existing PostgreSQL database is:

```
> dropdb $SPTOOL_DB
```

Once you either drop the existing database or change the assigned name, you can perform the initialization again.

### 3.4.3 Load the Speciation Tool *shared* schema data

The final step in initializing the Speciation Tool is to load the data that are provided in the distribution package. This information includes species properties, chemical mechanism definitions, speciation profile definitions from SPECIATE 4.3, and some additional general information that is shared by all Speciation Tool runs. The data are imported into the Speciation Tool database during initialization and stored in tables in the Speciation Tool PostgreSQL database *shared* schema.

A database contains one or more named schemas, which in turn contain tables and other objects. The same table name can be used in different schemas without conflict; for example, in Exhibit 2-1 the cb05_criteria and saprc_toxics schemas can both contain the same table names. Unlike PostgreSQL databases, schemas are not rigidly separated, so users can access tables in any of the schemas within the same database. The Speciation Tool *shared* schema is accessed by all Speciation Tool runs; the tables are shared.

The input data files reside in the /import_data subdirectory of the Speciation Tool home directory. The file names are assigned to environment variables in the *Assigns.sptool* file and do not need to be changed.

Messages are displayed during the initialization process to indicate progress while importing the shared data. Below is a partial sample of a successful load. The time it takes to import the data will vary depending on your system; a few minutes is typical.

```
Importing mechanisms  /disk4/models/emis/sptool/src/../import_data/mechanism_mar2013_forimport.txt

Using transactions - import of data will abort on any error.
Reading data from /disk4/models/emis/sptool/src/../import_data/mechanism_mar2013_forimport.txt...
  1000 lines processed
  2000 lines processed
…
  18000 lines processed
...finished, imported 18981 lines.

…

Importing species properties /disk4/models/emis/sptool/src/../import_data/export_species_properties_2013_0311.csv

Using transactions - import of data will abort on any error.
Reading data from /disk43/sptool/wa1_05.2012/src/../import_data/export_species_properties_2013_0311.csv...
  1000 lines processed
  2000 lines processed
...finished, imported 2274 lines.

Speciation Tool shared data successfully imported

Completed:  Wed Mar 20 11:50:30 PDT 2013
```

The Speciation Tool uses the *shared* schema functions and tables in every run. Appendix B provides the table definitions that are stored in the *shared* schema.

If an error occurs while importing the data, the program will terminate with a message indicating which file and record caused the error. A database rollback will be initiated and none of the data for the file with the error will be imported. The import stops at the first error. Use the printed message to diagnose the cause of the error. After correcting the error, rerun the load script.

### 3.4.4 The import program

The initialization script executes the Perl program *import_rawdata.pl* for each file imported to the *shared* schema. Each database table in the *shared* schema is associated with a keyword listed in Table 1 that identifies the table to populate. To import additional data to the *shared* schema tables, you can execute the Perl program from the command line. This will essentially append data to an existing table.

```
> perl $SPTOOL_SRC_HOME/import_rawdata.pl  $SPTOOL_DB  table_type  input_file
```

The parameters include the database name, a keyword from Table 3-1, and the path/file name of the records to use as input. The input data must conform to the table definitions in Appendix B. Any error will abort the import with a database rollback; no records will be appended to the table. Use the printed message to diagnose the cause of the error. After correcting the error, rerun the import script.

**Table 3-1. Keyword list showing Speciation Tool *shared* data types.**

| Keyword | Description |
|---|---|
| mechanism | mechanism definition |
| mechanismPM | $PM_{2.5}$ mechanism definition |
| mechanism_description | description and data source for each chemical mechanism |
| invtable | SMOKE Inventory Table |
| gas_profiles | gas profile description and historical information |
| gas_profile_weights | gas weight profiles |
| pm_profiles | PM profile description and historical information |
| pm_profile_weights | PM weight profiles |
| rename_species | rename mechanism AQM lumped species names |
| species | species data: IDs, names, molecular weights, etc. |
| carbons | number of carbons for AQM lumped species name |
| static | static profiles (profile weights do not change) |

After a successful Speciation Tool initialization, the PostgreSQL *sptoolv3_1* database will be created with the *shared* schema that holds the tool functions and default data.

# 4.0 RUNNING THE SPECIATION TOOL

A Speciation Tool run will generate the speciation factors required to convert emission inventory species (e.g., VOC or $PM_{2.5}$) to lumped species (e.g., OLE or ALD2) for air quality modeling. This chapter discusses the mechanics of running the Speciation Tool, which include how to run the tool and the parameters and options available.

The command to run the Speciation Tool is:

```
perl $SPTOOL_SRC_HOME/run_sptool.pl <database> <run_name> <run_control_file>
```

The command line parameters include the database name, user-specified run name, and control file name. The *run_sptool.pl* program creates the run schema, reads and imports the specified run control file, imports run-specific data files, executes the PostgreSQL functions to compute the split factors and conversion factors, and writes the output files.

Prior to running the Speciation Tool (discussed in Section 4.4), you will need to:

- Source the assigns file (Section 4.1)
- Create a run name (Section 4.2)
- Create a run control file (Section 4.3)
- Develop the input files that are required for the run (Section 4.3)

## 4.1 The Run Script

The Perl program *run_sptool.pl* creates the output speciation profile files GSPRO and GSCNV. The program requires the source code directory and database name. Source the *Assigns.sptool* file to set the required environment variables SPTOOL_HOME and SPTOOL_DB. This file was discussed in Section 3.4.1.

If you receive an error "file not found" when running the script, it is probably because you have not set the Speciation Tool home directory, or because the path is set to an invalid path name.

## 4.2 Run Name

The run name is an identifier to uniquely label the Speciation Tool run. It is used to create a schema of that name in the Speciation Tool database. To create a new schema without replacing an existing one, specify a run name that has not previously been used. If you specify an existing run name, the existing schema will be dropped and replaced by the new run.

The run name must meet the following conditions:

- Begin with an alphabetic character
- Must **not** include spaces in the name
- Must **not** be the *shared* schema
- Should be different from all existing run names in the Speciation Tool, unless you mean to replace an existing run

- Must **not** be "public" or begin with "pg_" (these are reserved for system schemas)
- Must be less than 64 characters long

To examine the list of existing schemas (runs) in the database, use this command:

```
psql   <database_name>   -c   'select * from pg_namespace'
```

where <database_name> is the Speciation Tool database name. The first column of the resulting list, labeled "nspname", will include run names as well as system schema names. The last column, labeled "nspacl", lists the owner of each schema.

## 4.3 Run Control File

The run control file indicates the run parameters and the run-specific data files required for a Speciation Tool run. The format of the run control file is

<keyword>, <option>

Only lines that begin with a keyword are recognized by the database. All other lines are skipped, allowing embedded comments to be used within the control file. Table 4-1 summarizes the control file keywords and their options. Details on each keyword are provided in Sections 4.3.1 through 4.3.3.

**Table 4-1. Speciation Tool Control File Keywords.**

| Keyword | Description |
|---|---|
| **Run Parameters** | |
| MECH_BASIS | Mechanism identifier/name.<br>Matches identifier in the shared schema mechanism table.<br>Options provided with Speciation Tool include:<br>• CBIV<br>• CB05<br>• CB6<br>• SAPRC99<br>• SAPRC99B<br>• SAPRC07<br>• SAPRC07T<br>• SOA_CAMX45<br>• SOA_CAMX45_SP99<br>• AE5<br>• AE6 |
| OUTPUT | Output options (default is VOC):<br>• VOC<br>• PM<br>• STATIC |
| RUN_TYPE | Specifies how TOG split factors are developed. Options include:<br>• CRITERIA<br>• INTEGRATE<br>• NOINTEGRATE<br>• HAPLIST |

| | |
|---|---|
| AQM | Air quality model. Determines some of the lumped species names for the output files. Options:<br>• CAMX<br>• CMAQ |
| TOLERANCE | The sum of weight percentages for a profile definition should equal 100%, but sometimes it does not. TOLERANCE specifies the acceptable deviation from 100%. The default is 5%. Applies only to gas profiles. |
| **Optional Input Files** | |
| PRO_FILE | User-defined input profiles path/file name. |
| TOX_FILE | Toxics species path/file name. |
| PRIMARY_FILE | Additional toxics entries path/file name. |
| PROC_FILE | Mobile-source process modes path/file name. |
| **Output Files** | |
| SPLITS_OUT | Output speciation profiles (GSPRO) path/file name. |
| CNV_OUT | Output conversion factors (GSCNV) path/file name. |

### 4.3.1 Run parameters

For all of the keywords below, only one option can be specified per run.

**MECH_BASIS** *Example:* MECH_BASIS, CB05

This keyword indicates the chemical mechanism to use for mapping inventory species to the AQM lumped species. The specified mechanism must match a mechanism name in one of the mechanism tables in the *shared* schema; these names are imported when the Speciation Tool is initialized. The data provided with the Speciation Tool include VOC mechanism definitions for CBIV, CB05, CB6, SAPRC07, SAPRC07T, SAPRC99, SAPRC99B, SOA_CAMX45, and SOA_CAMX45_SP99, and $PM_{2.5}$ mechanism definitions for AE5 and AE6.

**OUTPUT** *Example:* OUTPUT, VOC

The OUTPUT keyword indicates which output profiles will be generated. The output options are "VOC" (gas species), "PM" (particulates), and "STATIC". STATIC represents the list of splits that do not change with profile weight definitions, such as the pollutants CO and $NH_3$. It simply formats the imported static profiles to correspond to the output GSPRO file format.

The OUTPUT keyword is optional. If it is omitted from the control file, the default is to generate VOC factors.

**RUN_TYPE** *Example:* RUN_TYPE, CRITERIA

The RUN_TYPE options are CRITERIA, INTEGRATE, NOINTEGRATE, and HAPLIST. The different run types support different simulation options in SMOKE. *For VOCs output:* All four of these options can be used. For emissions processing of only total VOCs, the CRITERIA option is used. The INTEGRATE or NOINTEGRATE options are used if a separate toxics (HAP) inventory is included in addition to a total VOC inventory in the SMOKE modeling. HAPLIST supports the SMOKE calculation of NONHAPTOG. *For $PM_{2.5}$ output:* The only run type currently supported is CRITERIA.

- CRITERIA: All model species, including toxics species, are computed from criteria emissions. There is no separate toxics emission inventory included in the modeling.

- INTEGRATE: The HAPs species from a separate emission inventory are to be integrated. Part of the criteria VOC mass is replaced with the HAP VOC mass. This involves subtracting toxics VOC emissions from the criteria VOC emissions to avoid double counting of VOCs. Profiles are generated as NONHAPTOG.

- NOINTEGRATE: The HAPs species from a separate emission inventory are not to be integrated (i.e., toxics are included from both sources). This case assumes that the criteria VOC mass includes HAP VOC mass. The NOINTEGRATE approach is selected when sources in the criteria inventory and the toxics inventory do not have the necessary details to be definitively matched. The **active** HAPs (specified as explicit in the inventory table [INVTABLE]) are removed from the VOC profiles but the profiles are not renormalized, mass is preserved, and profiles are generated as TOG.

- HAPLIST: Generates records for the HAPs that define the SMOKE calculation of NONHAPTOG.

Refer to the SMOKE user's manual (http://www.smoke-model.org) for more detailed information on how SMOKE handles integrating criteria VOC and toxics inventories.

*Note:* The Speciation Tool *shared* tables includes a default INVTABLE file. These data should be reviewed and potentially replaced with the INVTABLE that you are using for SMOKE modeling.

### AQM *Example:* AQM, CAMX

The AQM keyword is used to indicate the air quality model to be used, so that the output files will contain appropriate model lumped species names. The Speciation Tool initialization data files support the AQM options CMAQ and CAMX. CMAQ is the Community Multiscale Air Quality Model, and CAMX is the Comprehensive Air Quality Model with extensions.

Different models sometimes use different names for the same species. For example, "other" $PM_{2.5}$ emissions are mapped to the model species name "PMFINE" in CMAQ but "FPRM" in CAMX. The tables tbl_static and tbl_rename_species in the shared schema include fields that indicate model-specific lumped species names. If another model uses different names, those need to be included in these tables. The *shared* schema table tbl_rename_species allows you to export model-specific species names different from the ones specified in the mechanism table. For example, ethene in the SAPRC99 mechanism is stored as "ETHE" in the mechanism table and is mapped to the model species name "ETHENE" for CMAQ.

### TOLERANCE *Example:* TOLERANCE, 3

Each profile is defined as the sum of its components and is quantified as the percent contribution of each component to total organic gases. Ideally, the sum of the percent components for each profile should equal 100%, which means that all mass is accounted for and assigned to individual species. However, due to round-off or inaccurate profile definitions, the profile components do not always add to 100%. The optional TOLERANCE keyword defines an acceptable deviation from 100%, with the default level in the Speciation Tool set at 5%. Any profile definition where the sum of the percentages is outside the acceptable tolerance will **not** be output in the run. For example, for the 5% default tolerance level, all profiles that sum to less than 95% or greater than 105% would be excluded from the output.

If the sum of the weight percentages for a particular profile is not 100% and nothing is done to correct the profile definition, the resulting speciation profile will drop or add mass if used in emissions modeling. The Speciation Tool therefore renormalizes all gas profiles whose sum is within the tolerance, thereby preserving VOC mass. Profiles outside of the tolerance are dropped at run time, although they do continue to reside in the *shared* schema tables.

### 4.3.2 Input files

File formats for all of the run-specific input files are provided in Appendix C.

**PRO_FILE** *Example:* PRO_FILE, ctl/prfwts_02coalstudy.dat

The PRO_FILE keyword is used to specify a file of gas profiles. Specifying this option will override using the gas profiles defined in the *shared* schema. The profiles provided in this input file are those for which the model will generate the splits and conversion factors. PRO_FILE is an optional keyword in the control file. The default is to use the profile definitions in the *shared* schema. At this time, only one set of profiles is used per run—either the *shared* data or the data provided using the PRO_FILE keyword. If speciation profiles are required from both sources, separate runs are required.

**TOX_FILE** *Example:* TOX_FILE, ctl/toxics_coalstudy.dat

The toxics file is applicable only to VOC output. This file is required for run types INTEGRATE, NOINTEGRATE, and HAPLIST. The file lists the toxics compounds that are explicit in the chemical mechanism. The Speciation Tool uses the information in the Inventory Table, imported to the *shared* schema during initialization, to determine whether each HAP is to be treated as an active or a tracer compound; an active compound is indicated with a "Y" in the INVTABLE *explicit* field. Active species are included in the chemical mechanism with chemical feedback, while tracer species are included with no chemical feedback. Tracer toxics species mass is double counted.

For the INTEGRATE case (i.e., RUN_TYPE, INTEGRATE), the active and tracer toxics species are removed from the VOC profiles, whereas for the NOINTEGRATE case, only active toxics species are removed from the VOC profiles.

For the case where HAPLIST is the specified RUN_TYPE, the active HAPs produce a single one-to-one mapping from HAPs inventory pollutant to model compound. The tracer HAPs generate the same mapping as the active HAPS, but in addition the tracer HAPs inventory pollutants are mapped to the original VOC profile compounds to support the INTEGRATE case. The HAPLIST case also generates "nointegrate" HAPs by appending the suffix "_NOI" to the model compound name (for example, BENZENE_NOI).

**PRIMARY_FILE** *Example:* PRIMARY_FILE, ctl/primary_toxics_coalstudy.dat

You can add or override toxics entries by providing a primary toxics list. This file is used only when the run type is HAPLIST; if a primary toxics list is specified for any other run type, the data are ignored. The file determines whether toxics species are added to or replaced in the GSPRO file. This capability allows the Speciation Tool to support one-to-many toxics species mapping and primary toxics profiles. The file format and example are as follows:

```
<Inv poll> <AQM name>   <splitfac> <overwrite flag>
FORMALD,   FORM_PRIMARY,   1.0,    N
ACETALD,   ALD2_PRIMARY,   1.0,    N
XYLS,      MXYL,           0.52,   Y
XYLS,      OXYL,           0.16,   Y
XYLS,      PXYL,           0.16,   Y
```

The overwrite flag in the last column determines whether a toxics species is added to ("N") or replaced in ("Y") the GSPRO file. In the input toxics table, XYLS (species ID = 507) has been assigned to DONT_USE (a placeholder). With the overwrite flag turned on, the Speciation Tool will remove the "XYLS DONT_USE" entry and add the three XYLS entries shown above. For the FORM_PRIMARY and ALD2_PRIMARY lumped species, on the other hand, the tool will add these species to GSPRO without removing any entries.

**PROC_FILE** *Example:* PROC_FILE, process_mode.dat

The PROC_FILE keyword specifies an optional input file that provides mobile-source emission modes for profiles that represent mobile-source processing. This feature is provided to support SMOKE requirements regarding mobile-source emission modes. An example record in this file is "4674, EXH" where profile code 4674 applies to exhaust emissions. When the process file is provided, additional records are generated in the GSCNV output file. For example, profile 4674 would include a VOC-to-TOG record as well as an EXH_VOC-to-EXH_TOG record. with the same conversion factors.

### 4.3.3 Output files

Keyword options

**SPLITS_OUT** *Example:* SPLITS_OUT, outputs/gspro_02coalstudy.dat

The SPLITS_OUT optional keyword is used to specify a path and file name for the model splits results written in SMOKE GSPRO format. Paths can be relative paths from the Speciation Tool main directory (as shown) or absolute paths. If this keyword is omitted, the output file name is derived from the run control parameter specifications and the run date and is written to the relative path /outputs. An example of a default output file name is outputs/gspro_CB05_NOINTEGRATE_CAMX_26Jun2011.

**CNV _OUT** *Example:* CNV_OUT, outputs/gscnv_02coalstudy.dat

The CNV_OUT optional keyword is used to specify a path and file name for the model TOG/VOC conversion factors written in SMOKE GSCNV format. Paths can be relative paths from the Speciation Tool main directory (as shown) or absolute paths. If this keyword is omitted, the output file name is derived from the run control parameter specifications and the run date and is written to the relative path /outputs. An example of a default output file name is outputs/gscnv_CB05_NOINTEGRATE_CAMX_26Jun2011.

Header Records: Metadata
The Speciation Tool output files, GSPRO and GSCNV, contain header records that are referred to as "metadata". Each of the output files has the same common metadata keywords. The purpose of the metadata records is to provide a summary of the data and parameters that were used to generate the outputs. Table 4-2 provides the list of metadata keywords for an example run.

**Table 4-2. Metadata keywords and example.**

| Metadata Keyword | Example |
|---|---|
| #SPTOOL_AQM | CMAQ |
| #SPTOOL_CARBONS | carbons_cb6_cb05_cbiv_saprc99.csv |
| #SPTOOL_GAS_PROFILES | Qry_gas_specie_export.txt |
| #SPTOOL_INVTABLE | invtable_caphap_27jun2007_v7.txt |
| #SPTOOL_MECH | CB05 |
| #SPTOOL_PM_PROFILES | Not Applicable |
| #SPTOOL_PROCESS | process_poll.csv |
| #SPTOOL_STATIC | Not Applicable |
| #SPTOOL_VOC_TYPE | INTEGRATE |

If the keyword is not applicable for a particular run, the corresponding value is set to "Not Applicable". For example, if the output type is specified as PM, the metadata record #SPTOOL_GAS_PROFILES is set to "Not Applicable". These headers can be used as a guide for merging different GSPRO tables. Before combining outputs for toxics TOG profiles with NONHAPTOG profiles, be sure to confirm that the same chemical mechanism was specified for each run and that the same input files for the VOC calculations were used for each run.

## 4.4 Run the Speciation Tool

To run the Speciation Tool, enter the following command:

```
perl $SPTOOL_SRC_HOME/run_sptool.pl $SPTOOL_DB <run_name> <run_control_file>
```

where:

$SPTOOL_DB is the Speciation Tool database name
<run_name> is the user-assigned run name
<run_control_file> is the name and location of the control file that defines the run parameters and input and output file names

### 4.4.1 Run-time messages

The Speciation Tool writes a number of messages to standard output that indicate the program's progress. The messages indicate key steps as they are completed either from the Perl scripts or the PostgreSQL functions. The difference between the message types is easily detected, as all of the PostgreSQL messages begin with the word "NOTICE". Exhibit 4-1 provides an example of a typical list of run-time messages for a run identified as "cb05_integrate". If the run name specified was used for a case run previously, additional messages are written at the start of the program that indicate existing tables are being dropped (e.g., "NOTICE: drop cascades to table cb05_integrate.tmp_metadataset").

```
Created cb05_integrate schema
Granted permissions on cb05_integrate schema
Set path successfully to cb05_integrate schema
Input tables created in cb05_integrate schema
Completed importing User Profile Weights file user_prof_wts_2540.txt
Imported run control file run_ctl_cb05_integrate.txt in cb05_integrate schema
NOTICE:  Type of Output is  VOC
NOTICE:  AQM is  CAMX
NOTICE:  Type of run is  INTEGRATE
NOTICE:  Mechanism basis is  CB05
NOTICE:  Profile Tolerance is  5
NOTICE:  ...establishing profile weights
NOTICE:  ...renormalizing profile weights
NOTICE:  ...establishing mechanism
NOTICE:  ...calculating moles per gram emissions
NOTICE:  ...calculating mole percent
NOTICE:  ...calculating moles per mole emissions
NOTICE:  ...summing on AQM pollutant
NOTICE:  ...calculating mole weight percent
NOTICE:  ...calculating average molecular weight by specie
NOTICE:  ...calculating average molecular weight by AQM
Completed splits calculations
Completed output preparations
Completed writing the output files for run cb05_integrate
```

**Exhibit 4-1. Example of run-time messages.**

There are a number of different warning messages that are displayed during processing. The most frequent message occurs when a species exists in the *shared* schema species table that has not been defined in the specified chemical mechanism. Below is a small example of these messages, which provide the species ID number and name. There are many species that are currently undefined in the chemical mechanisms; this is a concern only if a profile references one of them.

```
NOTICE:  WARNING review vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
NOTICE:  WARNING: CB05 mechanism undefined for the following species.
NOTICE:        The following will be set to UNK.
NOTICE:  WARNING:        SPECIES ID  2673 2,6-dimethylheptane, propylcyclopentane
NOTICE:  WARNING:        SPECIES ID  2674 Trans-3-hexene; 3-methylcyclopentene
NOTICE:  WARNING:        SPECIES ID  2675 2-methyl-2-hexene; cis-3-heptene
NOTICE:  WARNING:        SPECIES ID  2676 3-methyl-trans-3-hexene; Trans-2-heptene
```

As indicated in Section 4.3.1, profiles are dropped during processing when the weight percent sum is outside the specified tolerance. The dropped profiles are reported to standard output. Below is a small example of these warnings.

```
NOTICE:  ...establishing profile weights
NOTICE:  WARNING review vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
NOTICE:  WARNING The following profiles were dropped.
NOTICE:  WARNING Total weight percent is outside the tolerance of 5 percent.
NOTICE:  WARNING:   PROFILE ID  4742 82.87586700 percent Diesel Exhaust - Bus at -10 oC, 2-stroke
NOTICE:  WARNING:   PROFILE ID  4743 83.00138200 percent Diesel Exhaust - Bus at 20 oC, 2-stroke
NOTICE:  WARNING:   PROFILE ID  4744 84.50961700 percent Diesel Exhaust - Bus at -10 oC, 2-stroke, oxidatio
```

Note that many of these listed profiles actually have a profile definition with a percent sum of 100%. The problem is not with the profile definition but the incomplete species data. Any species with an undefined/missing molecular weight is dropped from the profile, since the molecular weight is required to compute the splits factors. This dropped percent sometimes takes the total profile percent sum outside the selected tolerance. This type of problem can be resolved by updating the species' molecular weights.

### 4.4.2 What happens when the script is running?

The *run_sptool.pl* program creates the run schema, reads and imports the run control file, imports the run-specific data files, executes the PostgreSQL functions that compute split factors and conversion factors, and writes the output files. After the script has been run, the run-specific schema retains the run parameters, input data, intermediate calculations, and final results. Appendix C includes the run-based schema table formats for the input data and a list of the database tables created during a run.

The Speciation Tool is driven by the Perl program *run_sptool.pl*. Table 4-3 summarizes the program steps. In addition to the commands below, numerous data and processing checks are performed in the run script.

**Table 4-3. Program and module calls from the Speciation Tool program run_sptool.pl.**

| Command line | Language | Description |
|---|---|---|
| SELECT inputs_createtables | SQL | Executes PostgreSQL commands in table_inps.sql:<br>• Creates tables in the run-id schema. |
| import_control.pl | Perl | Imports the run control data and optional user-specified input data. |

| | | |
|---|---|---|
| SELECT MakeSplits | SQL | If VOC processing:<br>Executes PostgreSQL commands in make_splits.sql:<br>• Creates tables in the run_id schema.<br>• Performs data validity checks.<br>• Performs numerous calculations to generate the gas split factors. |
| SELECT MakePMSplits | SQL | If $PM_{2.5}$ processing:<br>Executes PostgreSQL commands in make_pm_splits.sql:<br>• Creates tables in the run_id schema.<br>• Performs data validity checks.<br>• Performs calculations to generate the $PM_{2.5}$ split factors. |
| SELECT PrepOut | SQL | Executes PostgreSQL commands in prep_out.sql:<br>• Creates tables in the run_id schema.<br>• Generates the output splits and conversion factors data. |
| write_outputs.pl | Perl | Extracts from the run_id schema the split factors and conversion factors data and writes the output GSPRO and GSCNV files. |

# 5.0 APPLICATIONS AND METHODOLOGY

## 5.1 Gas Profile Processing

The Speciation Tool is designed to support the availability of both CAPS and HAPS; to either integrate the HAPs in the chemical mechanism or not; and to include both active and tracer species in modeling. It generates GSPRO and GSCNV speciation input files for the SMOKE model. Refer to the SMOKE user's manual for detailed information of how SMOKE handles integrating the criteria VOC and toxics inventories.

The Speciation Tool *shared* schema includes the table *tbl_invtable,* which carries the imported default INVTABLE data. These data should be reviewed to verify that they correspond to the SMOKE INVTABLE that will be used in your SMOKE modeling.

### 5.1.1 Run type options

As discussed in Section 4.3.1, the various Speciation Tool run options are CRITERIA, INTEGRATE, NOINTEGRATE, and HAPLIST. CRITERIA is specified if no additional HAPS inventory is included in the modeling. The INTEGRATE and NOINTEGRATE options indicate how a separate HAPS inventory will be handled in the SMOKE model, by either subtracting the HAPS portion from the criteria VOC or not. The HAPLIST option generates the HAPS records used in SMOKE for computing NONHAPVOC.

CRITERIA
A run type of CRITERIA means that all VOC model species, including toxics VOC species, are computed from criteria VOC emissions. This option is specified when there is no separate toxics emission inventory included in the modeling.

INTEGRATE
A run type of INTEGRATE means that the specified HAPs species are integrated from a separate emission inventory. Part of the criteria VOC mass will be replaced with the HAP VOC mass from the HAP emission inventory. SMOKE computes the NONHAPVOC mass from the criteria VOC mass by subtracting the HAP VOC mass.

The Speciation Tool creates the HAPs list using the Inventory Table data that are imported during initialization. The HAPs list is composed of species/compounds in the Inventory Table where the field Keep = Y **AND** the field VOCTOG = (V **OR** T). The HAPs are removed from the VOC profiles and the profiles are renormalized and generated as NONHAPVOC profiles. The GSCNV output file will contain NONHAPVOC-to-NONHAPTOG factors.

The GSPRO header records list all of the toxics species that are subtracted from TOG to estimate NONHAPTOG. If the Inventory Table includes process mode information for on-road mobile-source modeling, some pollutant names will also be appended to the process mode. For example,

```
#NHAP NONHAPTOG BENZENE
#NHAP NONHAPTOG EVP__BENZENE
```

SMOKE compares the INVTABLE settings to the header records in the GSPRO file, ensuring that the lists are consistent.

NOI**N**TEGRATE
In the NOINTEGRATE case, both the criteria VOC inventory and the separate toxics inventory are included in emissions processing. However, due to an inability to map one inventory to the other, toxics are included from both sources. The criteria VOC includes the HAP VOC mass. In generating the speciation profiles, the specified **active** HAPs are removed from the VOC profiles but the profiles are not renormalized, mass is preserved, and profiles are generated as TOG.

HAPLIST
This option generates records for the HAPs that define the NONHAPTOG for both the INTEGRATE and NOINTEGRATE cases. All HAPS are written to profile number "0000", which is often used as the default profile ID in SMOKE processing.

### 5.1.2 Chemical mechanisms

During initialization, the Speciation Tool imports default data (profiles, species list, chemical mechanism, etc.) into the *shared* schema. However, the Speciation Tool is not tailored to a specific chemical mechanism (such as CB05). If the chemical mechanism you require is not available in the Speciation Tool database, you can import a new mechanism definition, assuming you can supply the new chemical mechanism assignments. Refer to Section 6.3 in the Chapter 6 FAQ list for these details.

The Speciation Tool also has the capability to generate speciation profiles for secondary organic aerosols (SOA) to support air quality models such as CAMx that have an explicit SOA scheme. This is accomplished through the use of the chemical mechanism SOA_CAMX45, which is included in the initialization data.

When using the SAPRC99 mechanism, keep in mind that the carbon content of VOC or TOG emissions is not conserved through the SAPRC99 speciation step, that the output mass fractions in the GSPRO file are sensitive to the number of carbons assumed for each species, and that carbon numbers are not well-defined for SAPRC99 lumped species. This uncertainty does not influence the output modeling emissions in molar units. Output emissions reported in mass units (such as speciated SMOKE reports), however, do inherit this uncertainty and therefore should be considered as qualitative rather than quantitative information.

## 5.2 $PM_{2.5}$ Profile Processing

The Speciation Tool also supports speciation profile entries for $PM_{2.5}$. At this time only the CRITERIA run option is supported for PM processing. The chemical mechanism definitions imported during initialization include the CMAQ aerosol mechanisms AE5 and AE6. Appendix D provides these mechanism definitions.

The $PM_{2.5}$ chemical mechanism definitions include a list of species that define the mechanism, a flag ("qualify") to define the species that determine the profiles to process, and the AQM pollutant name to be assigned to any unspecified PM mass (e.g., PMFINE).

A profile is processed if it contains one or more species that are flagged "qualify". For each profile that matches the selection criteria, only the species that are specifically defined in the chemical mechanism definition will be included in the outputs. Each $PM_{2.5}$ chemical mechanism definition must have a single record specifying the AQM pollutant to compute; it does not matter whether the specified computed pollutant is a component of the profile definition. The unspecified mass component is defined as 1.0 minus the sum of the mass fractions of the mechanism compounds. In the AE5 chemical mechanism, the unspecified mass is assigned to PMFINE, and in AE6 it is assigned to the output species PMOTHR.

# 6.0 FREQUENTLY ASKED QUESTIONS

## 6.1 How do I add a single speciation profile?

The easiest method of processing one or more new profiles is to use the run control keyword PRO_FILE. The profile definition is imported to a run schema table. If this table is populated, only profiles in the run schema are processed.

## 6.2 Can I have more than one copy of the Speciation Tool database?

Yes. If you want to rerun the Speciation Tool initialization and keep an already existing instance of the Speciation Tool database, simply make a copy of the *Assigns.sptool* file and change the database name variable SPTOOL_DB. "Source" the new assigns file and run the initialization program. A new database is created and the data files imported.

This is the approach to take if you want to modify any of the shared data. In particular, you may have different versions of the INVTABLE that you want to support. Each version can be represented in a different database.

## 6.3 How do I define a new chemical mechanism?

To introduce a new chemical mechanism to the Speciation Tool:

- Define the assignments to the mechanism table with a unique mechanism name (different from those of existing mechanisms) and import the data to the shared schema mechanism table (tbl_mechanism for VOC processing, tbl_pm_mechanism for $PM_{2.5}$ processing).

- If needed, update the *shared* schema carbon table (tbl_carbons) to include any new model species names.

- Update the mechanism description table (tbl_mechanism_description) in the *shared* schema.

For examples of the tables used to define new chemical mechanisms, look under the import_data directory in the Speciation Tool distribution package for the following files:

tbl_mechanism: mechanism_14may2013_forimport.txt
tbl_carbons: carbons_14may2013.csv
tbl_mechanism_description: mechanism_description.14may2013.txt

Refer to Appendix B for the file formats.

The new mechanism data can be imported with the *import_rawdata.pl* Perl program:

```
perl  $SPTOOL_SRC_HOME/import_rawdata.pl $SPTOOL_DB  table_type  input_file
```

where:

table_type is the keyword from Table 1 (in this example it would be either mechanism, carbons, or mechanism_description)

input_file contains the specified new records

Alternatively, rather than appending records to three separate shared schema tables (tbl_mechanism, tbl_carbons, tbl_mechanism_description), a new version of the Speciation Tool could be generated (refer to question 6.2 above). The new mechanism definition data could be imported as part of the initialization process. These data could be appended to existing files or not, depending on your processing preferences.

The example below illustrates how to introduce a new mechanism, "SAPRC99 with explicit benzene".

1. Update the mechanism table:

   Extract the SAPRC99 entries from the import file:

   ```
   grep ^SAPRC99 mechanism_mar2013_forimport.txt > mechanism_saprc99_wexpl_benz.txt
   ```

   Edit the new file:

   Change SAPRC99 to SAPRC99_BENZ (on all records)
   Delete existing benzene (specie_id = 302) assignments of "ARO1" and "NROG"
   Add explicit benzene entry "SAPRC99_BENZ, 302, BENZ, 1"

2. Import the new mechanism file to the shared mechanism table:

   ```
   perl import_rawdata.pl database_name mechanism mechanism_saprc99_wexpl_benz.txt
   ```

3. Update the carbon table:

   Create a file with the record "SAPRC99_BENZ,BENZ,6"

   Import the file to the shared carbon table:

   ```
   perl import_rawdata.pl database_name carbons <new_carbon_file.txt>
   ```

4. Update the mechanism description table:

   Create a file with the record "SAPRC99_BENZ, SAPRC99 with explicit benzene, Y, <description>,,"

   Import the file to the shared mechanism description table:

   ```
   perl import_rawdata.pl database_name mechanism_description <new_file.txt>
   ```

The mechanism description table carries the field "NONSOA Flag", which defines whether the mechanism treats SOA explicitly. The CBIV, CB05, and SAPRC99 mechanisms all have the NONSOA flag set to "Y". Only the SOA_CAMX45 mechanism has the flag set to "N".

## 6.4 What if I have a different INVTABLE?

The SMOKE INVTABLE data is imported as part of the Speciation Tool initialization. Currently the tool does not allow an optional input of inventory data for a single run. If you need to use only your version of the INVTABLE, replace the INVTABLE file name that is referenced in the *Assigns.sptool* before you initialize the database.

If you need to use multiple versions for your processing applications, you will need multiple versions of the Speciation Tool database, each importing a different INVTABLE during initialization (refer to question 6.2 above).

## 6.5 How do I add a new species to the database?

To add a single record from the command line, use an INSERT statement. The *shared* schema *tbl_species* carries almost 20 fields (exported from the SPECIATE database). The Speciation Tool references only a few of these fields. The required fields are specie_id, specie_name, volatile_mw, and non_voctog. Here is an example of adding a single record to the table with only the required fields:

```
psql –c “INSERT INTO shared.tbl_species (specie_id,specie_name,volatile_mw,non_voctog)
                                 VALUES (‘EG1’, ‘Example Name’, 70.273, FALSE);
```

The specie_id must be unique in the table; you will get an error message if you attempt to add an already existing *specie_id*. The *non_voctog* field is type Boolean and should be set to true if the species is not regarded as a VOC.

# APPENDIX A

## PostgreSQL and Perl Installation Procedures

This section provides details on installing PostgreSQL and the required Perl modules (DBI, DBD-Pg, and Text-CVS) on a Linux system via the Linux command line. Before continuing, it is recommended that you discuss these software requirements with your system administrator, as you may need to be logged in as `root`.

## A.1. Status of Requirements

Begin by determining what is and is not installed on your system. Execute the script *sptool_reqd_checks.sh,* which is provided in the Speciation Tool package. The script checks for the required software and user PostgreSQL authorizations.

```
> ./sptool_reqd_checks.sh
```

Following is an example of a successful check for the required software:

```
===== Speciation Tool Requirements Check =====

Checking the status of software requirements...

Status of required software:
[x] = Installed
[ ] = Not installed
[?] = Unable to determine, see notes

------PERL------
[x] Perl
[x]    -DBI
[x]    -DBD-Pg
[x]    -Text-CSV

---POSTGRESQL---
[x] PostgreSQL
[x]    -PL/pgSQL

Refer to the Speciation Tool User Guide Appendix A for installation procedures of the required software.
```

Missing software/modules are indicated by a `[ ]`, whereas software/modules that are installed and ready to be used are marked with `[x]`. Note that if PostgreSQL is erroneously shown as not installed, restarting the postgresql service can fix many of the reasons that might have caused this to occur.

## A.2. Perl

For the purposes of this guide, it is assumed that Perl has already been installed on your system; almost every modern Linux distribution includes Perl. To verify that Perl is installed, enter the following at the Linux command line:

```
# perl -v
```

If Perl is present, information about the version will be displayed. Additional interfaces and modules for Perl need to be installed that will allow the Speciation Tool to read text files and communicate with the database; installation instructions are given in Section A.4. First, however, PostgreSQL must be installed, as described in Section A.3.

You can use the YUM software package manager to install PostgreSQL and the Perl modules. YUM is an easy-to-use utility that installs, updates, and removes software packages on RPM-based systems. This utility is found on RedHat, CentOS, Fedora, and other RPM-based Linux distributions.

*Note:* For systems with different installation procedures/requirements, PostgreSQL precompiled binaries can be found at http://www.postgresql.org/. Click on the Downloads tab to review the available products. From http://www.postgresql.org/download/, choose the binary package corresponding to your operating system. As of August 2013, binary packages are available for the following operating systems:

- BSD
  - FreeBSD
  - OpenBSD
- Linux
  - RedHat/CentOS/Fedora/Scientific families Linux
  - Debian GNU/Linux
  - Ubuntu Linux
  - SuSE and OpenSuSE
  - Other Linux
- Mac OS X
- Solaris
- Windows

## A.3. PostgreSQL Installation

To install PostgreSQL using YUM, you will probably need to be logged in as `root`. To download and install PostgreSQL, execute the following command:

```
# yum install postgresql
```

If prompted, enter "y" to confirm the installation. The package will download and install; additional packages may be automatically selected for install in order to resolve dependencies.

## A.4. Install Perl Database Interface Modules

Three Perl modules are required to read in text files and communicate with the database used by the Speciation Tool:

| | |
|---|---|
| Perl DBI<br>(perl-DBI) | The Perl Database Interface allows communication between the Speciation Tool scripts and PostgreSQL database. |
| Perl DBD-Pg<br>(perl-DBD-Pg)<br>*or* Perl DBD-PgPP | This is a PostgreSQL-specific database driver for the DBI module to allow Perl to communicate with the PostgreSQL database using the DBI.<br>Note that for this release of the Speciation Tool, the DBD-PgPP driver (instead of DBD-Pg) was used for the installation and testing. |
| Perl Text-CSV<br>(perl-Text-CSV) | This module allows Perl scripts to parse and create CSV (comma separated values) files. |

Use YUM to download and install (if needed) all three modules. Enter the following command (note that the package names are case sensitive):

```
# yum install perl-DBI perl-DBD-Pg perl-Text-CSV
```

Enter "y" when prompted to confirm the installation of these packages.

*Note:* If the system displays errors indicating that the packages are not found, check the spelling and case of the package names in your command.

As an alternative to YUM, you can use the CPAN installation approach with Perl. Enter the following commands:

```
# perl –MCPAN –e shell
cpan> install DBI
cpan> install Text::CSV
cpan> install DBD::PgPP
```

## A.5. Start the PostgreSQL Service and Prepare It for the Speciation Tool

To start the PostgreSQL database service, execute the following command:

```
# /etc/init.d/postgresql start
```

Change to the user who will be running the Speciation Tool:

```
# su <user name here>
```

Start the terminal-based front end to PostgreSQL (psql) with the default postgres user:

```
> psql –U postgres
```

Create PostgreSQL user(s) with the database create option for each of the users who will be running the Speciation Tool. Each user name should correspond to the Linux user account name. The command is:

```
=# CREATE USER <user name here> WITH CREATEDB;
```

Then exit psql:

```
        =# \q
```

Finally, run the script *sptool_reqd_checks.sh* again (as in Section A.1) to verify that all of the packages are available to the Speciation Tool. If they are, you are now ready to follow the Speciation Tool setup procedures in Chapter 3.

# APPENDIX B

## *Shared* Schema File Formats

### Table B-1. tbl_carbons.

| Field Name | Field Type | Description |
|---|---|---|
| mechanism | Character 20 | Mechanism name |
| aqm_poll | Character 20 | Air quality model lumped species identifier/name |
| num_carbons | Numeric (5,2) | Number of carbon bonds for lumped species |

### Table B-2. tbl_gas_profile_weights.

| Field Name | Field Type | Description |
|---|---|---|
| profile_id | Character 20 | Profile identifier |
| specie_id | Character 20 | Unique species identifier |
| percent | Numeric (10,6) | Percentage of pollutant in profile |
| uncertainty | Numeric (10,6) | Uncertainty percent of pollutant |
| unc_method | Character 100 | Description of method used to determine uncertainty |
| analytic_method | Character 500 | Description of analytical method |

### Table B-3. tbl_gas_profiles.

| Field Name | Field Type | Description |
|---|---|---|
| profile_id | Character 20 | Unique profile identifier |
| profile_name | Character 200 | Profile name |
| quality | Character 10 | Overall Quality Rating |
| controls | Character 100 | Emission controls |
| date_added | Date | Date profile added |
| notes | Character | Notes |
| total | Numeric (10,5) | Sum of profile percentages |
| master_poll | Character 20 | Basis of profile |
| test_method | Character | Test method description |
| norm_basis | Character 100 | Normalization description |
| composite | Character 1 | Indicates if profile is original or composite; O or C |
| standard | Boolean | Standard or user added profile |
| test_year | Character 50 | Year testing was completed |
| j_rating | Numeric (10,2) | Judgment rating based on general merit |
| v_rating | Numeric (10,2) | Vintage Rating |
| d_rating | Numeric (10,2) | Data quality rating based on number of observations |
| region | Character 100 | Geographic region of relevance |
| old_profile | Integer | Profiles taken from previous version |
| sibling | Character 20 | Profile_ID of PM profile from same study |
| voc_to_tog | Numeric (12,7) | VOC to TOG conversion factor |
| data_origin | Character 50 | Originating organization |
| primary_prof | Boolean | |
| description | Character | Profile description |
| documentation | Character | Documentation of profile origin |

### Table B-4. tbl_invtable.

| Field Name | Field Type | Description |
|---|---|---|
| eminv_poll | Character 12 | Emission inventory pollutant name |
| mode | Character 3 | Process mode |
| poll_code | Character 16 | Pollutant code |
| specie_id | Character 20 | Species identifier |
| reactivity | Character 20 | Reactivity group |
| keep | Character 20 | Keep flag; Y or N |
| factor | Character 20 | Adjustment factor |
| voc | Character 20 | VOC or TOG component flag |
| model | Character 20 | Model species flag |
| explicit | Character 20 | Explicit in mechanism flag |
| activity | Character 20 | Data type flag; Y indicates activity (not emissions) |
| nti | Character 20 | Identifies HAPs on the Clean Air List |
| unit | Character 20 | Units |
| description | Character 50 | Inventory data description |
| cas_description | Character 50 | CAS pollutant description |

### Table B-5. tbl_mechanism.

| Field Name | Field Type | Description |
|---|---|---|
| mechanism | Character 20 | Mechanism name |
| specie_id | Character 20 | Unique species identifier |
| aqm_poll | Character 20 | Air Quality Modeling lumped species identifier |
| moles_per_mole | Numeric (20,12) | The moles per mole |

### Table B-6. tbl_mechanism_description.

| Field Name | Field Type | Description |
|---|---|---|
| mechanism | Character 20 | Mechanism name |
| description | Character 256 | Description |
| nonsoaflag | Character 1 | Flag: "N" SOA mechanism<br>"Y" nonSOA mechanism |
| origin | Character 300 | Originating organization |
| reference | Character 100 | References |
| comment | Character 500 | Comment |

### Table B-7. tbl_metadata.

| Field Name | Field Type | Description |
|---|---|---|
| keyword | Character 20 | Metadata keyword |
| datival | Character 256 | Corresponding value (file) |
| version | Character 20 | Data version |

## Table B-8. tbl_pm_mechanism.

| Field Name | Field Type | Description |
|---|---|---|
| mechanism | Character 20 | Mechanism name |
| specie_id | Character 20 | Unique species identifier |
| aqm_poll | Character 20 | Air Quality Modeling lumped species identifier |
| qualify | Boolean | Indicates which species determines if a profile qualifies for the specified mechanism |
| compute | Boolean | If true, compute aqm_poll (usually PMFINE or PMOTHR) |

## Table B-9. tbl_pm_profiles.

| Field Name | Field Type | Description |
|---|---|---|
| profile_id | Character 20 | Unique profile identifier |
| profile_name | Character 255 | Profile name |
| quality | Character 10 | Overall Quality Rating |
| controls | Character 150 | Emission controls |
| date_added | Date | Date profile added |
| notes | Character | Notes |
| total | Numeric (10,5) | Sum of profile percentages |
| master_poll | Character 20 | Basis of profile |
| test_method | Character | Test method description |
| norm_basis | Character 100 | Normalization description |
| composite | Character 1 | Indicates if profile is original or composite; O or C |
| standard | Boolean | Standard or user added profile |
| incl_gas | Boolean | |
| test_year | Character 50 | Year testing was completed |
| j_rating | Numeric (10,2) | Judgment rating based on general merit |
| v_rating | Numeric (10,2) | Vintage Rating |
| d_rating | Numeric (10,2) | Data quality rating based on number of observations |
| region | Character 100 | Geographic region of relevance |
| lower_size | Numeric (10,4) | |
| upper_size | Numeric (10,4) | |
| sibling | Character 20 | Profile_ID of PM profile from same study |
| data_origin | Character 50 | Originating organization |
| primary_prof | Boolean | |
| description | Character | Profile description |
| documentation | Character | Documentation of profile origin |

## Table B-10. tbl_pm_profile_weights.

| Field Name | Field Type | Description |
|---|---|---|
| profile_id | Character 20 | Profile identifier |
| specie_id | Character 20 | Unique species identifier |
| percent | Numeric (12,6) | Percentage of pollutant in profile |
| uncertainty | Numeric (12,6) | Uncertainty percent of pollutant |
| unc_method | Character 100 | Description of method used to determine uncertainty |
| analytic_method | Character 500 | Description of analytical method |

### Table B-11. tbl_rename_species

| Field Name | Field Type | Description |
|---|---|---|
| aq_model | Character 10 | AQM Model |
| mechanism | Character 20 | Mechanism name |
| eminv_poll | Character 20 | Compound name |
| aqm_poll | Character 20 | Replacement AQM compound name |

### Table B-12. tbl_species.

| Field Name | Field Type | Description |
|---|---|---|
| specie_id | Character 20 | Unique species identifier |
| specie_name | Character 100 | Species name |
| CAS | Character 50 | CAS identifier |
| epaid | Character 50 | EPA identifier |
| saroad | Character 10 | Old SAROAD code |
| pams | Boolean | Yes or No to PAMS |
| haps | Boolean | Yes or No to HAPs |
| symbol | Character 10 | Symbolic name |
| molecular_weight | Numeric (20,12) | Molecular weight of specie |
| non_voctog | Boolean | Yes or No to non-volatile organic gas |
| non_vol_wt | Character 20 | Non-volatile weight |
| unknown_wt | Character 20 | Unknown weight |
| unassign_wt | Character 20 | Unassigned weight |
| exempt_wt | Character 20 | Exempt weight |
| volatile_mw | Numeric (20,12) | Volatile molecular weight |
| num_carbons | Numeric (20,12) | Number of carbon bonds |
| epa_itn | Character 20 | EPA internal tracking number |
| comment | Character 50 | Comments |

### Table B-13. tbl_static.

| Field Name | Field Type | Description |
|---|---|---|
| profile_id | Character 20 | Profile identifier |
| eminv_poll | Character 20 | Emission inventory pollutant code |
| aqm_poll | Character 20 | AQM species code |
| split_factor | Numeric (20,10) | Mole based split factor (numerator) |
| divisor | Numeric (20,10) | Denominator of the mole based factor |
| mass_fraction | Numeric (20,10) | Mass fraction |
| aq_model | Character 10 | Air Quality Model |

# APPENDIX C

## Run Schema File Formats

**Table C-1. tbl_gas_process.**

| Field Name | Field Type | Description |
|---|---|---|
| profile_id | Character 20 | Profile identifier |
| process | Character 20 | Process mode |

**Table C-2. tbl_primary.**

| Field Name | Field Type | Description |
|---|---|---|
| aqminv_poll | Character 20 | Inventory pollutant name |
| aqm_add | Character 20 | Compound name to add |
| split_factor | Numeric (12,8) | Split factor for added compound |
| writeflag | Character 1 | Flag: "N" – compound should be added<br>"Y" – compound should be replaced |

**Table C-3. tbl_run_control.**

| Field Name | Field Type | Description |
|---|---|---|
| keyword | Character 20 | Key word identifying function or file name |
| datival | Character | Corresponding value for the key word |

**Table C-4. tbl_toxics.**

| Field Name | Field Type | Description |
|---|---|---|
| aqm_model | Character 20 | AQM name |
| specie_id | Character 20 | Unique species identifier |
| aqm_poll | Character20 | Air quality model lumped species identifier/name |
| num_carbons* | Numeric (6,3) | Number of carbon bonds for lumped AQM species |
| active* | Character 1 | Options: A – active, T - tracer |

* Not a user input; assigned by the system.

**Table C-5. tbl_user_profile_wts.**

| Field Name | Field Type | Description |
|---|---|---|
| profile_id | Character 20 | Profile identifier |
| specie_id | Character 20 | Unique species identifier |
| percent | Numeric (10,6) | Percentage of pollutant in profile |

### Table C-6. Table names in the run schema created by the Speciation Tool for internal calculations.

| Table Name | Description |
|---|---|
| tmp_actox | Checks for profiles with 100% active toxics. |
| tmp_aqm_carbons | Initialized with shared .tbl_carbons. Records are added for user specified toxics species. |
| tmp_calcs_byaqm | Store intermediate calculations by mechanism, profileId, and AQM pollutant. Includes moles per gram, moles model species per mole emissions, and average molecular weight . |
| tmp_calcs_byspc | Stores intermediate calculations by mechanism, profileId, specieId, and AQM pollutant. Includes moles per gram, moles model species per mole emissions, mole weight percent of model species, grams per mole, and average molecular weight. |
| tmp_calcs_haps | Store intermediate calculations by mechanism, profileId, specieId, emission inventory pollutant, and AQM pollutant for the user specified HAPs species. Includes moles per mole, moles per gram, and average molecular weight. |
| tmp_calcs_haps_null | Stores the calculated average molecular weights for those records in tmp_calcs_haps with undefined average molecular weights. |
| tmp_error | Stores run error messages. |
| tmp_gscnv | Stores the output conversion factor data. |
| tmp_gspro | Stores the output splits factor data. |
| tmp_haps | Stores the list of HAPs extracted from the INVTABLE. |
| tmp_header | Stores the NONHAPTOG header entries. |
| tmp_invtable | Used for checking output pollutant name widths specified in INVTABLE. |
| tmp_mechanism | Initialized with the user selected mechanism from shared tbl_mechanism or tbl_pm_mechanism. Records are removed or inserted depending on run parameters. |
| tmp_metadataset | Stores the metadata written to the header of the output files. |
| tmp_prfwts | Stores the |
| tmp_pm_mechanism | Mechanism definition for user specified PM mechanism run |
| tmp_pm_splits | $PM_{2.5}$ split factors |
| tmp_prfwts | Initialized with tmp_raw_profiles. Data records are removed and renormalized depending upon run parameters. |
| tmp_profile_list | Stores the unique list of PM profiles |
| tmp_qa_carbons | Stores AQM compounds where no carbon data has been specified. |
| tmp_qa_mechanism | Stores species with no corresponding mechanism definition. |
| tmp_raw_profiles | Initialized with either shared .tbl_gas_profile_weights, tbl_pm_profile_weights, or tbl_user_profile_wts if user provided. |
| tmp_species_carbons | Stores the calculated number of carbons for each species. |
| tmp_spcinp | Stores user specified species that have invalid molecular weights. |
| tmp_sums | Stores the profile weights percent sums during renormalization. |

# APPENDIX D

## $PM_{2.5}$ Chemical Mechanism Definitions

Table D-1 represents the AE5 and AE6 chemical mechanism definitions. Table D2, prepared by EPA, provided the information used to fill in Table D1.

**Table D-1. Chemical Mechanisms AE5 and AE6 as input to the Speciation Tool.**

| Chemical Mechanism | SpecieID | AQM Pollutant | Qualify | Compute |
|---|---|---|---|---|
| **AE5** | | | | |
| AE5 | 626 | POC | T | F |
| AE5 | 797 | PEC | T | F |
| AE5 | 699 | PSO4 | T | F |
| AE5 | 613 | PNO3 | T | F |
| AE5 | | PMFINE | F | T |
| **AE6** | | | | |
| AE6 | 626 | POC | F | F |
| AE6 | 797 | PEC | F | F |
| AE6 | 699 | PSO4 | F | F |
| AE6 | 613 | PNO3 | F | F |
| AE6 | 784 | PNH4 | F | F |
| AE6 | 2669 | PNCOM | T | F |
| AE6 | 488 | PFE | F | F |
| AE6 | 292 | PAL | F | F |
| AE6 | 694 | PSI | F | F |
| AE6 | 715 | PTI | F | F |
| AE6 | 329 | PCA | F | F |
| AE6 | 525 | PMG | F | F |
| AE6 | 669 | PK | F | F |
| AE6 | 526 | PMN | F | F |
| AE6 | 696 | PNA | F | F |
| AE6 | 795 | PCL | F | F |
| AE6 | 2668 | PH2O | T | F |
| AE6 | | PMOTHR | F | T |

## Table D-2. Chemical Mechanisms AE5 and AE6 as defined by EPA

| SPECIATE speciesID | Species Description | Species Name | Calculation | Notes |
|---|---|---|---|---|
| | CMAQ AE5: | | | |
| 626 | organic carbon | POC | explicit | measured |
| 797 | elemental carbon | PEC | explicit | measured |
| 699 | sulfate | PSO4 | explicit | measured |
| 613 | nitrate | PNO3 | explicit | measured |
| | unspeciated $PM_{2.5}$ | PMFINE | 1-(sum of 4 species) | n/a |
| | CMAQ AE6: | | | |
| 626 | organic carbon | POC | explicit | measured |
| 797 | elemental carbon | PEC | explicit | measured |
| 699 | sulfate | PSO4 | explicit | measured |
| 613 | nitrate | PNO3 | explicit | measured |
| 784 | ammonium | PNH4 | explicit in 911XX profiles | measured |
| 2669 | non-carbon organic matter | PNCOM | explicit in 911XX profiles | PNCOM = POC*(OM/OC Ratio - 1)<br>where OM/OC ratio is 1.25 for motor vehicle exhaust, 1.7 for wood combustion, 1.4 for other sources. |
| 488 | iron | PFE | explicit in 911XX profiles | measured |
| 292 | aluminum | PAL | explicit in 911XX profiles | measured |
| 694 | silica | PSI | explicit in 911XX profiles | measured |
| 715 | titanium | PTI | explicit in 911XX profiles | measured |
| 329 | calcium | PCA | explicit in 911XX profiles | measured |
| 525 | magnesium | PMG | explicit in 911XX profiles | measured |
| 669 | potassium | PK | explicit in 911XX profiles | measured |
| 526 | manganese | PMN | explicit in 911XX profiles | measured |
| 696 | sodium | PNA | explicit in 911XX profiles | measured |
| 795 | chloride | PCL | explicit in 911XX profiles | measured |
| 2668 | water | PH2O | explicit in 911XX profiles | 0.24*(PNH4+PSO4) for non-combustion sources, 0 for combustion sources or use measured value of hydrated water, if available |
| | unspeciated $PM_{2.5}$ | PMOTHR | 1 - (sum of 17 species) | n/a |

* A note on chloride from M. Strum (EPA OAQPS): only species 795 (chlorine atom) is used to populate PCL rather than using species 337 (chloride ion) or using both 795 and 337 because in the current $PM_{2.5}$ profiles in SPECIATE 4.3 when chloride (species 337) is present species 795 (chlorine atom) is also present. Including both of these would be double counting. In addition, per communication with Adam Reff (email from Adam Reff to Madeleine Strum, 3/24/2011), Reff indicates that chlorine atom (via XRF) is a more complete representation of all forms of Cl that might be present.