SMOKE v4.0 User's Manual

SMOKE v4.0 User's Manual

Published $Date: 2016/9/30

Copyright © 2016 The institute for the Environment - The University of North Carolina at Chapel Hill

Table of Contents

Chapter 1. Introduction

Table of Contents

1.1. Synopsis of SMOKE

The environmental community has developed advanced numerical air quality models (AQMs) to understand the interactions among meteorology, emissions (both manmade and biogenic), and pollutant chemistry and dynamics. Emissions data from emissions models and regulatory inventories are one of the most important inputs for these air quality models. Scientists use air quality modeling for a number of purposes: for state and federal implementation plan development, for research on improved modeling methods, and most recently for air quality forecasting. In all of these cases, the trend has been to model larger regions, at a finer grid resolution, with more emissions sources, and for more purposes (e.g., ozone, particulates, toxics). These needs require a computationally efficient, user-friendly, and flexible emissions data processing system.

The MCNC Environmental Modeling Center (EMC) created the Sparse Matrix Operator Kernel Emissions (SMOKE) Modeling System to allow emissions data processing methods to integrate high-performance-computing (HPC) sparse-matrix algorithms. The SMOKE system is a significant addition to the available resources for decision-making about emissions controls for both urban and regional applications. It provides a mechanism for preparing specialized inputs for air quality modeling research, and it makes air quality forecasting possible. The SMOKE system continues to develop and improve at the University of North Carolina at Chapel Hill's Institute for the Environment (IE).

The SMOKE prototype, available since 1996, was an effective tool for emissions processing in a number of regional air quality modeling applications. In 1998 and 1999, SMOKE was redesigned and improved with the support of the U.S. Environmental Protection Agency (EPA). The primary purposes of the first SMOKE redesign were support of (1) emissions processing with user-selected chemical mechanisms (described in more detail in Section 2.11, "Chemical speciation processing" [70]) and (2) emissions processing for reactivity assessments (described in Section 2.14.3, "Creating the reactivity control matrix" [81]). In 2002, SMOKE was enhanced to support driving the MOBILE6 model used to create on-road mobile emission factors and to support on-road and nonroad mobile toxics inventories, resulting in SMOKE v1.5. In 2003, SMOKE v2.0 was created to include all toxic inventories, including point and nonpoint (stationary sources reported at the county level) sources. SMOKE v2.1, released in 2004, included updated versions of the BEIS3 and MOBILE6.2 models, the ability to use humidity data when processing on-road mobile sources with MOBILE6, and support for polar stereographic output grids. In 2009, SMOKE v2.6 enhanced the processing of fire data, streamlined the processing for CAMx, REMSAD, and UAM models, and added a new approach to processing CEM data. Version 2.7, SMOKE was enhanced to support processing the MOVES (MOter Vehicle Emission Simulator) emission rates output through the SMOKE system to model on-roadway and off-network mobile sources. As of version 4.0 SMOKE is enhanced to support hemispheric modeling by enabling the processing of global gridded emission inventories, such as EDGAR, for input to chemistry-tranport models.

SMOKE can process criteria gaseous pollutants such as carbon monoxide (CO), nitrogen oxides (NO_x), volatile organic compounds (VOC), ammonia (NH_3), sulfur dioxide (SO_2); particulate matter (PM) pollutants such as PM 2.5 microns or less ($PM_{2.5}$) and PM less than 10 microns (PM_{10}); as well as a large array of

toxic pollutants, such as mercury, cadmium, benzene, and formaldehyde. In fact, SMOKE has no limitation regarding the number or types of pollutants it can process.

The purpose of SMOKE (or any emissions processor) is to convert the resolution of the emission inventory data to the resolution needed by an air quality model. Emission inventories are typically available with an annual-total emissions value for each emissions source, or perhaps with an average-day emissions value. The AQMs, however, typically require emissions data on an hourly basis, for each model grid cell (and perhaps model layer), and for each model species. (Refer to Appendix A. Glossary [499] for definitions of these terms.) Consequently, emissions processing involves transforming an emission inventory through temporal allocation, chemical speciation, and spatial allocation, to achieve the input requirements of the AQM.

1.2. SMOKE Users

Possible SMOKE users are individuals who need to prepare emission input files for one of the following air quality models:

- Community Multiscale Air Quality (CMAQ) modeling system
- Multiscale Air Quality Simulation Platform (MAQSIP)
- Regulatory Modeling System for Aerosols and Deposition (REMSAD)
- Comprehensive Air Quality Model, with Extensions (CAM$_X$)
- Urban Airshed Model, version 4 (UAM-IV)
- Urban Airshed Model, Variable grid (UAM-V)

We anticipate that the typical SMOKE user will have some combination of the following experience and needs:

1. Those knowledgeable about emission inventories.
2. Those with UNIX experience (including Linux).
3. Those with little or no emissions modeling background.
4. Those who are somewhat familiar with grid-based air quality models and their emissions input needs.
5. Persons with emission inventories in hand who need to process the data for input to an AQM.
6. EPA personnel who want to create emission inputs for research or regulatory efforts with AQMs.
7. State environmental personnel who want to create emission inputs to AQMs for regulatory efforts, including State Implementation Plans (SIPs).
8. Those wishing to generate an on-road mobile inventory using MOVES and either the Weather Research and Forecasting (WRF) modeling system or the Pennsylvania State University/National Center for Atmospheric Research fifth-generation Mesoscale Model (MM5) gridded, hourly meteorology data.

SMOKE users who have previous UNIX experience can expect to run SMOKE more easily than users who must learn both UNIX and SMOKE. This is because the standard way to run SMOKE is from UNIX scripts.

1.3. How to Use this Manual

This document is the most complete reference available on SMOKE. It is available both as a set of HTML pages and PDF document (including page numbers) from the CMAS center [http://www.cmascenter.org].

Given below are typical activities for which users need this manual, and the resources available to meet those needs.

- **Install SMOKE:** Installation instructions are available in Chapter 12, *Downloading, Installing, and Compiling SMOKE* [495]. To register and download SMOKE, visit the CMAS Model Clearinghouse [http://www.cmascenter.org/html/models.html].

- **Compile SMOKE:** Section 12.4, "Compiling SMOKE for UNIX" [496] provides instructions for compiling SMOKE for platforms other than the default ones supported by the CMAS Center.

- **Run SMOKE test case:** Section 4.3, "Running the SMOKE test case" [124] provides instructions for running the test case provided with SMOKE.

- **Set up and run SMOKE for new inventory, grid, episode, etc.:** Section 4.4, "How to use SMOKE" [125] includes instructions on how users can set up SMOKE for their own cases. The SMOKE classroom training [http://www.cmascenter.org/training.cfm] available through the CMAS Center covers these topics in more detail.

- **Prepare input files:** Most SMOKE input files are ASCII files that can be prepared using database programs or other tools. Descriptions of all input file formats are found in Chapter 8, *SMOKE Input Files* [345]. Users may obtain input data files (perhaps not in SMOKE input format) from EPA web sites, such as the Clearinghouse for Inventories and Emission Factors (CHIEF) web site [http://www.epa.gov/ttn/chief/emch/].

- **Decide how to assign settings in run scripts:** In Chapter 4, *Using SMOKE Scripts* [117], we cover the SMOKE scripts and how to use them. The settings are assigned via environment variables, which are explained in general terms in Section 2.2, "Assigns file and environment variables" [6]. Any settings used to control the scripts (not the programs) are described in Section 4.2.4, "Script settings" [121]. The settings that are evaluated by SMOKE programs and directly affect their behavior are further explained in Chapter 5, *SMOKE Utility Tools* [177], Chapter 6, *SMOKE Core Programs* [211], and Chapter 7, *SMOKE Quality Assurance* [305].

- **Check that SMOKE has run properly:** Section 7.5, "Quality Assurance Methods" [335] includes a list of steps regarding how to verify that SMOKE has run properly. This chapter also includes instructions on running the quality assurance (QA) features of SMOKE and documentation on the reporting control file (Section 7.3, "REPCONFIG Input File" [308]) needed for configuring different types of reports for quality assurance.

- **Understand the operations that SMOKE is performing:** Any available technical documentation regarding SMOKE operations is provided in Chapter 5, *SMOKE Utility Tools* [177], Chapter 6, *SMOKE Core Programs* [211], and Chapter 7, *SMOKE Quality Assurance* [305] for each SMOKE program. We are continually working to enhance this documentation and can take suggestions through the CMAS Center Help Desk (see next bullet).

- **Submit problem reports or questions about SMOKE:** All questions and problems concerning SMOKE and other Models-3 components should be submitted through the CMAS Center Help Desk [http://www.cmascenter.org/html/help.html].

This manual does not provide substantial technical documentation on the code itself, though some information is provided in Chapter 11, *Source Code and Include Files* [491]. The code does have extensive in-line documentation to assist Fortran-savvy users in understanding its workings. We hope that in the future we will be able to enhance the documentation in Chapter 11, *Source Code and Include Files* [491], but are unsure of the necessity and have not obtained resources for such documentation to date.

Chapter 2. SMOKE Concepts

Table of Contents

2.1. Introduction

The purpose of SMOKE is to convert the resolution of the data in an emission inventory to the resolution needed by an air quality model. Emission inventories typically have an annual-total emissions value for each emissions source, or perhaps an average-day emissions value. The AQMs, however, typically require emissions data on an hourly basis, for each model grid cell (and perhaps model layer), and for each model species. Consequently, to achieve the input requirements of the AQM, emissions processing must (at a minimum) transform inventory data by temporal allocation, chemical speciation, spatial allocation, and perhaps layer assignment.

In addition to changing the resolution of the data, SMOKE must also provide the AQM input files in the correct file format. SMOKE can create the Input/Output Applications Programming Interface (I/O API) Network Common Data Form (NetCDF) output format needed by the CMAQ and MAQSIP models. It can also create the Fortran binary format for the 2-D emissions needed by REMSAD, UAM, and CAM$_X$, and the ASCII elevated-point-source format used by the Ptsrce preprocessor to these models. File format is also important for the input files used by SMOKE, most of which are ASCII files, but some of which are I/O API NetCDF or CF-compliant NetCDF format files.

In this chapter, we introduce you to various concepts that are critical to understanding the technical description of emissions processing, as well as provide more detail about the processing capabilities of SMOKE. (Later, Chapter 5, *SMOKE Utility Tools* [177], Chapter 6, *SMOKE Core Programs* [211], Chapter 7, *SMOKE Quality Assurance* [305], Chapter 8, *SMOKE Input Files* [345], Chapter 9, *SMOKE Intermediate Files* [461], and Chapter 10, *SMOKE Output Files* [483] give more specifics about each program's capabilities and each file's format.) This chapter provides the context and framework for the rest of the user's manual. To assist

you in reading and using this chapter, we provide Appendix A. Glossary [499] for definitions of emissions inventory and emissions modeling terminology.

2.2. Assigns file and environment variables

The Assigns file is a script used to set up the parameters of a SMOKE run. The file configures the UNIX environment so that all of the correct input, intermediate, and output directories and files can be identified and used by the SMOKE programs. It also sets things like the name of the grid and the time period that you will run SMOKE for a given case. It does this by setting many UNIX environment variables, explained in the next paragraph. The Assigns file also uses environment variables to configure compiler options, so that SMOKE can be compiled on operating system other than the ones provided with the SMOKE distribution. More information on the Assigns file is provided in Section 4.2.2.2, "Assigns files" [120].

Environment variables are aliases that can be set by a UNIX operating system. These variables are defined during a user's UNIX session, usually defined by an **xterm** or other UNIX terminal window. The environment variables that SMOKE uses store the input, intermediate, and output files and directories. For example, the environment variable for the directory that is the SMOKE root directory is SMKROOT. At the UNIX prompt, this environment variable could be defined to an actual path such as /home/mylogin/smoke. To set an environment variable, the UNIX **setenv** command is needed. In this example, the command to define SMKROOT as the given path is:

setenv SMKROOT /home/mylogin/smoke

After this command is issued, the SMKROOT environment variable stores the characters /home/mylogin/smoke as its value. To use the value of an environment variable, the dollar sign must proceed the variable name at the UNIX prompt. In the follow example, we give the UNIX command **echo** to print the contents of the SMKROOT environment variable at the UNIX prompt. Note the use of the dollar sign before the SMKROOT variable name.

echo $SMKROOT

When the UNIX system executes this command, the following is displayed at the UNIX prompt:

/home/mylogin/smoke

The environment variables set by the Assigns file for directories are described in Chapter 3, *SMOKE Directory Structure* [97]. The variables used by the SMOKE scripts for controlling SMOKE execution are described in Section 4.2.4, "Script settings" [121]. Finally, the environment variables that control program behavior are described in Chapter 5, *SMOKE Utility Tools* [177], Chapter 6, *SMOKE Core Programs* [211], and Chapter 7, *SMOKE Quality Assurance* [305].

2.3. Emission inventories

Emission inventories are the key input files to SMOKE and emissions modeling. The data types that these inventories contain are called inventory pollutants (e.g., carbon monoxide, ammonia, mercury). By itself, SMOKE does not require specific data types in the inventory files it reads. However, the AQMs that SMOKE supports do require certain input data, called model species, which in turn requires SMOKE to use certain inventory pollutants.

In this section, we focus on the inventory files that SMOKE uses. Section 2.3.1, "Inventory data types" [7] describes the major inventory types useable by SMOKE. In Section 2.3.2, "Inventory source categories" [7], we describe the inventory source categories, and in Section 2.3.3, "Inventory file formats" [13] we discuss the inventory file formats. The remaining sections describe the various codes used in specific inventory

sources: Section 2.3.4, "Country, state, and county codes" [13], Section 2.3.5, "Source Classification Codes" [14], Section 2.3.6, "Standard Industrial Classification codes" [15], Section 2.3.8, "Maximum Achievable Control Technology codes" [16], Section 2.3.9, "Source types: major and section-112 area sources" [16] and Section 2.3.10, "Source types: nonroad and onroad mobile sources" [17].

2.3.1. Inventory data types

SMOKE processes criteria, particulate, toxics, and activity data inventories. Activity data will be discussed along with on-road mobile sources in the next section. By criteria inventories, we mean inventories containing EPA's criteria pollutants: carbon monoxide (CO), nitrogen oxides (NO_x), and volatile organic compounds (VOC) or total organic gases (TOG). Particulate inventories contain ammonia (NH_3), sulfur dioxide (SO_2), particulate matter (PM) of size 10 microns or less (PM_{10}), and PM of size 2.5 microns or less ($PM_{2.5}$).

Additionally, SMOKE can process inventories with pre-speciated criteria and/or particulate emissions. For example, elemental carbon of size 2.5 microns or less can be provided as input to SMOKE directly, instead of letting SMOKE's speciation step compute it from the $PM_{2.5}$ total emissions. To ensure that SMOKE correctly processes the data when you are using pre-speciated emissions, other input files must be configured in specific ways, as explained in Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150].

The toxics inventories that SMOKE can process are data from the National Emission Inventory (NEI) for Hazardous Air Pollutants (HAPs). This inventory contains hundreds of specific compounds representing the 188 HAPs defined by the Clear Air Act. The original list of 189 HAPs and modifications representing the current list are available from the EPA's web site [http://www.epa.gov/ttn/atw/orig189.html]. The reason the inventory contains many more pollutants than 188 is because several on the list of 188 are pollutant groups, such as polycyclic organic matter, cyanide compounds and numerous metal compounds including chromium compounds, cadmium compounds, manganese compounds, and others. Note that because of these groups, specific compounds in the inventory in one inventory year may not exactly match the compounds in another inventory year. For example, one may have lead oxide reported one year but not in a subsequent year. However, those compounds not belonging to compound groups are likely to be in the inventory year after year, particularly the common gaseous HAPs emitted by mobile sources such as benzene, 1,3-butadiene, acrolein, formaldehyde, and acetaldehyde.

2.3.2. Inventory source categories

2.3.2.1. Overview

Emission inventories are divided into several source categories. These divisions stem from both differing methods for preparing the inventories and from different characteristics and attributes of the categories (more on these terms later). Generally, emission inventories are divided into the following source categories:

- **Stationary area/Nonpoint sources:** Sources that are treated as being spread over a spatial extent (usually a county or air district) and that are not moveable (as compared to nonroad mobile and onroad mobile sources). Because it is not possible to collect the emissions at each point of emission, they are estimated over larger regions. The EPA introduced the term "nonpoint" to replace "stationary area" in order to avoid confusion with the term "area source", which is used as a regulatory term in the toxics realm. However, "nonpoint" has not gained acceptance (thus far) by the criteria inventory/modeling community. Thus, in this manual we will use the term "stationary area" to refer to these sources when they are in criteria inventories, while we use the term "nonpoint" to refer to these sources when they are in toxics inventories. Examples of nonpoint or stationary area sources are residential heating and architectural coatings. Numerous sources, such as dry cleaning facilities, may be treated either as stationary area/nonpoint sources or as point sources; in particular, the toxics inventory contains numerous

small sources (based on emissions) that are not inventoried as nonpoint sources because their locations are known and are provided.

- **Nonroad mobile sources:** Vehicular and otherwise movable sources that do not include vehicles that travel on roadways. These sources are also computed as being spread over a spatial extent (again, a county or air district). Examples of nonroad mobile sources include locomotives, lawn and garden equipment, construction vehicles, and boating emissions. These sources are included in both criteria and toxics inventories.

- **On-road mobile sources:** Vehicular sources that travel on roadways. These sources can be computed either as being spread over a spatial extent or as being assigned to a line location (called a link). Data in on-road inventories can be either emissions or activity data. Activity data consists of vehicle miles traveled (VMT) and, optionally, vehicle speed. Activity data are used when SMOKE will be computing emission factors via another model such as MOVES. Examples of on-road mobile sources include light-duty gasoline vehicles and heavy-duty diesel vehicles. On-road mobile sources are included in both criteria and toxics inventories.

- **Point sources:** These are sources that are identified by point locations, typically because they are regulated and their locations are available in regulatory reports. Point sources are often further subdivided into electric generating utilities (EGUs) and non-EGU sources, particularly in criteria inventories in which EGUs are a primary source of NO_x and SO_2. Examples of non-EGU point sources include chemical manufacturers and furniture refinishers. Point sources are included in both the criteria and toxics inventories.

- **Wildfire sources:** Traditionally, wildfire emissions have been treated as stationary area sources. More recently, data have also been developed for point locations, with day-specific emissions and hour-specific plume rise (vertical distribution of emissions). In this case, the wildfire emissions are processed by SMOKE as point sources.

- **Biogenic land use data:** Biogenic land use data characterize the type of vegetation that exists in either county total or grid cell values. The biogenic land use data in North American is available using two different sets of land use categories: the Biogenic Emissions Landcover Database (BELD) version 2 (BELD2), and the BELD version 3 (BELD3).

Emission processing in SMOKE is divided into four processing categories: area, biogenic, mobile, and point. The definitions of these categories that SMOKE uses are different than those used for defining emission inventories. Table 2.1, "Inventory source categories and SMOKE processing capabilities and categories" [9] lists the inventory source categories, the types of inventories (activity data, criteria, particulates, and toxics) that SMOKE can process, the temporal resolution that is acceptable to SMOKE, and the SMOKE processing category that should be used for processing the inventory.

Table 2.1. Inventory source categories and SMOKE processing capabilities and categories

| Inventory source category | Temporal resolution that SMOKE can process* | | | | SMOKE processing category |
	Activity data	Criteria	Particulates	Toxics	
Nonpoint or stationary area	N/A	A, S, D, H	A, S, D, H	A, S, D, H	Area
Nonroad mobile	N/A	A, S, D, H	A, S, D, H	A, S, D, H	Area
On-road mobile (MOBILE 6)	A	A, S, D, H	A, S, D, H	A, S, D, H	Mobile
On-road mobile (MOVES)	A	H	H	H	Mobile
EGU	N/A	A, S, D, H	A, S, D, H	A, S, D, H	Point
Non-EGU	N/A	A, S, D, H	A, S, D, H	A, S, D, H	Point
Wildfire with precomputed plume rise	N/A	D, H	D, H	N/A	Point
Wildfire with internal plume rise calculation	N/A	D	D	N/A	Point
Biogenic land use	N/A	X	N/A	N/A	Biogenic
* **A** = Supports annual data; **S** = Supports average-day data; **D** = Supports day-specific data; **H** = Supports hourly data; **X** = Supports available data					

2.3.2.2. Detailed source category descriptions

Each inventory source category has source characteristics, source attributes, data values, and data attributes. *Source characteristics* are unique to each inventory source category and also distinguish one source in the inventory from another. *Source attributes* further describe the sources with other information that is useful for emissions processing, such as point-source flue gas exit height and temperature. The *data values* are either emissions values or activity values. The *data attributes* are additional information about the data values, such as the percentage reduction in emission from controls already applied to the source. In the following subsections, we summarize the source characteristics and attributes and the data values and attributes that are used by SMOKE for each of the inventory categories.

2.3.2.2.1. Nonpoint/stationary area and nonroad mobile (SMOKE category: area)

- **Source characteristics:** For all typical inventories, the source characteristics that identify these sources are country/state/county code, SCC and/or GEOCODE_LEVEL[1-4]. See Section 2.3.4, "Country, state, and county codes" [13] and Section 2.3.5, "Source Classification Codes" [14] and Section 2.3.7, "Geographical Code (GEOCODE_LEVEL[1-4]" [15] for further information.

- **Optional source characteristics:** SMOKE can also use pregridded data from the same modeling domain as a SMOKE area source; this is described in more detail in Section 2.8.2.2, "" [32]. In this case, the source characteristics and attributes, (country/state/county code and SCC) are *not* used in SMOKE. SMOKE can also use pregridded data from a different modeling domain along with geographical codes (GEOCODE_LEVEL[1-4]) and source information from the ARINV file to specify the source characteristics and associated source attributes for each grid cell.

- **Source attributes:** The inventory year is associated with all sources in the inventory input files. In addition, SMOKE assigns a time zone (see Section 2.9.14, "Assign country codes, years, and time zones" [60]) and an approach for normalization of temporal profiles (see Section 2.9.13, "Set the weekday averaging approach" [60]). In the nonpoint toxics inventory only, Standard Industrial Classification (SIC) codes, Maximum Achievable Control Technology (MACT) codes, and North American

Industrial Classification System (NAICS) codes are optional source attributes; the NAICS code is read by SMOKE but not otherwise used at this time. Additionally, a "source type" field is available in the nonpoint inventory to identify major and Clean Air Act (CAA) section 112 area sources. See Section 2.3.6, "Standard Industrial Classification codes" [15] for a description of SIC codes, Section 2.3.8, "Maximum Achievable Control Technology codes" [16] for more about MACT codes, and Section 2.3.9, "Source types: major and section-112 area sources" [16] for more about source types. We will refer to the CAA section 112 area sources as simply "section-112 area sources".

- **Data:** SMOKE can read emissions data for criteria, particulate, and toxics pollutants for nonpoint/stationary area and nonroad inventories. The SMOKE system is not constrained with regard to the pollutants read (although typical examples were given in Section 2.3.1, "Inventory data types" [7]). SMOKE accepts annual emissions data, average-day emissions data, or both (though not all input formats support all types). An emission factor value can also be read by SMOKE, but SMOKE does nothing with it.

- **Data attributes:** Inventories for nonpoint/stationary area and nonroad mobile sources can contain control efficiency, rule penetration, and rule effectiveness information for each pollutant. SMOKE will use these data if provided; otherwise it will set default values that indicate that no control-based adjustments have been applied to the inventory pollutant data. The defaults are listed in the file formats in Chapter 8, *SMOKE Input Files* [345].

2.3.2.2.2. On-road mobile (SMOKE category: mobile)

- **Source characteristics:** For on-road mobile inventories, the minimum source characteristics that identify these sources are country/state/county code and either SCC *or* road class and vehicle type codes. When the SCC is provided, it must follow a specific pattern in order to contain the road class and vehicle type codes (see Section 2.8.4.1, "Special approach for on-road mobile MOVES SCCs" [37]). When road class and vehicle type codes are provided to SMOKE directly, SMOKE translates these to SCC values.

- **Optional source characteristics:** A link code may also identify on-road sources. This code must be unique within each county and SCC (or road class/vehicle type combination).

- **Source attributes:** The inventory year is associated with all sources in the inventory input files. In addition, SMOKE assigns a time zone (see Section 2.9.14, "Assign country codes, years, and time zones" [60]) and an approach for normalization of temporal profiles (see Section 2.9.13, "Set the weekday averaging approach" [60]). For sources with link codes, SMOKE will use the starting and ending coordinates of the link, using either latitude-longitude (lat-lon) values or coordinates in the Universal Transverse Mercator (UTM) coordinate system.

- **Data:** Emissions data for criteria, particulate, and toxics pollutants can be read for on-road mobile inventories. SMOKE is not constrained with regard to the pollutants read (although typical examples were given in Section 2.3.1, "Inventory data types" [7]). SMOKE accepts annual emissions data, average-day emissions data, or both (though not all input formats support all types).

  Additionally, on-road mobile inventories can contain VMT and average speed activity data, which are needed when users would like SMOKE to run MOVES to compute emissions. A combination of pre-computed emissions and VMT data is also acceptable for input to SMOKE, but you are responsible for preventing duplication of emissions. Duplication could occur if you input precomputed emissions for the same sources that you use SMOKE to compute the emissions on the fly, by multiplying the on-road emissions factors from MOVES by hourly VMT, and the off-network emission factors from MOVES by annual vehicle populations.

- **Data attributes:** No data attributes are associated with on-road mobile sources.

2.3.2.2.3. Point sources (SMOKE category: point)

- **Source characteristics:** The source characteristics for point sources depend on the inventory input format. The SMOKE one-record-per-line (ORL) and Flat File 10 Format (FF10) formats identify sources by country/state/county code, plant code, point code, stack code, segment code, and SCC. 95.

- **Optional source characteristics:** SMOKE can support up to five location identifiers within a plant, although the most used in any currently implemented input file format is four.

- **Source attributes:** As with other source categories, inventory year is associated with all sources in the inventory input files. SMOKE also assigns a time zone (see Section 2.9.14, "Assign country codes, years, and time zones" [60]) and an approach for normalization of temporal profiles (see Section 2.9.13, "Set the weekday averaging approach" [60]). In addition, point sources have the following required source attributes not associated with other source categories: latitude, longitude, stack height, stack diameter (at the exit location), flue gas exit velocity, and flue gas exit temperature. Finally, the following optional source attributes are also used by SMOKE: SIC codes, MACT codes, plant descriptions, emissions release type point (e.g., horizontal stack, fugitive), source type (major or section-112 area), Office of Regulatory Information Systems (ORIS) identification codes, and boiler identification codes. Also, the MACT code and source types are supported only by the ORL format. See Section 2.3.6, "Standard Industrial Classification codes" [15], Section 2.3.8, "Maximum Achievable Control Technology codes" [16], and Section 2.3.9, "Source types: major and section-112 area sources" [16] for more information.

- **Data:** Emissions data for criteria, particulate, and toxics pollutants can be read for point inventories. SMOKE is not constrained with regard to the pollutants read (although typical examples were given in Section 2.3.1, "Inventory data types" [7]). SMOKE accepts annual emissions data, average-day emissions data, or both.

  Optionally, point-source emissions data can be provided using day-specific or hour-specific records. The formats for these data are described in Section 8.2.6, "`PTDAY`: Point source day-specific emissions" [367] and Section 8.2.7, "`PTHOUR`: Point source hour-specific emissions" [369].

- **Data attributes:** EGU and non-EGU point sources can contain control efficiency and rule effectiveness information for each pollutant. SMOKE will use these data if provided; otherwise it will set default values that indicate that no control-based adjustments have been applied to the inventory pollutant data. The defaults are listed in the file formats in Chapter 8, *SMOKE Input Files* [345].

2.3.2.2.4. Wildfire with precomputed plume rise (SMOKE category: point)

- **Source characteristics:** Wildfires with precomputed plumes are identified by the country/state/county code and the fire name.

- **Optional source characteristics:** There are no optional source characteristics for wildfire sources.

- **Source attributes:** Like other source categories, inventory year is associated with all sources in the inventory input files. SMOKE also assigns a time zone (see Section 2.9.14, "Assign country codes, years, and time zones" [60]) and an approach for normalization of temporal profiles (see Section 2.9.13, "Set the weekday averaging approach" [60]). In addition, wildfire sources require the latitude and longitude source attributes. Finally, additional hour-specific source attributes for wildfire sources *must* be provided for the fraction of emissions in the surface layer, the height of the bottom of the plume, and the height of the top of the plume. These hour-specific attributes are provided to SMOKE using the point source hour-specific formats described in Section 8.2.7, "`PTHOUR`: Point source hour-specific emissions" [369].

- **Data:** Wildfire source inventories can contain criteria and particulate pollutants. SMOKE is not constrained with regard to the pollutants read (although typical examples were given in Section 2.3.1, "Inventory data types" [7]). These data must be provided as day-specific or hour-specific emissions values using point source formats specified in Section 8.2.6, "PTDAY: Point source day-specific emissions" [367] and Section 8.2.7, "PTHOUR: Point source hour-specific emissions" [369].

- **Data attributes:** No data attributes are associated with wildfire sources.

2.3.2.2.5. Wildfires with internal plume rise calculation (SMOKE category: point)

- **Source characteristics:** Wildfires with internal plume rise calculation are identified by the country/state/county code, fire identification, fire name, location identification, and SCC.

- **Optional source characteristics:** There are optional source characteristics for fire sources, such as material burned, vegetation types, size of area burned, fuel loading, and fire start/end hour. The size of area burned and fuel loading are used for computing the fire-specific plume rise. Fire starting and ending hours are needed to adjust the hourly temporal profiles for the emissions.

- **Source attributes:** Like other source categories, inventory year is associated with all sources in the inventory input files. SMOKE also assigns a time zone (see Section 2.9.14, "Assign country codes, years, and time zones" [60]) and will re-normalize temporal profiles based on the starting and ending hours of the fire. In addition, wildfire sources require the latitude and longitude source attributes to locate the fire. Note that all emissions for a fire will be assumed to come from the single grid cell that contains the latitude and longitude of the fire. Finally, additional day-specific source attributes listed above for fire sources *must* be provided for calculating the heat flux of each fire, which is used to estimate the fraction of emissions in the surface layer, the height of the bottom of the plume, and the height of the top of the plume. See more information about how to process at Section 4.4.17, "Plume Rise Calculation for Fires" [167]

- **Data:** Fire source inventories can contain criteria and particulate pollutants. SMOKE is not constrained with regard to the pollutants read (although typical examples were given in Section 8.2.8.3, "ORL FIRE Format" [380]). These data must be provided as day-specific emissions values using point source formats specified in Section 8.2.6.2, "ORL FIRE Emission Format" [368].

- **Data attributes:** No data attributes are associated with wildfire sources.

2.3.2.2.6. Biogenic land use (SMOKE category: biogenic)

- **Source characteristics:** Biogenic emission data does not fit as neatly into the source-characteristic paradigm as the previously described source types. Emissions for biogenic sources are estimated starting with land use data, which are available for both BELD2 and BELD3 processing. The BELD2 data are available either by U.S. state/county and BELD2 land use category or by grid cell and BELD2 land use category. BELD3 land use data are available by 1-km grid cell over North and Central America and by BELD3 land use category.

- **Optional source characteristics:** Biogenic land use data do not include optional source characteristics. The data are either by state/county or by grid cell.

- **Source attributes:** There are no source attributes for biogenic land use data.

- **Data:** The biogenic land use data consist of fractions associated with each land use type within a county or grid cell.

- **Data attributes:** There are no data attributes for biogenic land use data.

2.3.3. Inventory file formats

SMOKE supports a variety of inventory formats for criteria, particulate, toxics, and activity data inventories, which are described in detail in Section 8.2, "Inventory Files" [351]. Here, we provide a brief introduction to these formats, which will be helpful as you read more about SMOKE in the remainder of this chapter and the chapters before Chapter 8, *SMOKE Input Files* [345]. All formats described here are text files. To convert your data to these formats, the best approach is to use a database or spreadsheet program to reformat and output the data in the requested format. There is not a standard format-conversion method that comes with SMOKE.

In the following paragraphs, we describe the formats available for nonpoint/stationary area, nonroad mobile, on-road mobile, point, and point-wildfire sources.

- **Nonpoint/stationary area sources:** SMOKE supports two formats for nonpoint/stationary area sources. The ORL and FF10 (Flat File 10) format are list directed (comma or semicolon delimited) and these file formats may be used to represent many different sources. The header of the file indicates what source data are in the file.

- **Nonroad mobile sources:** There are three available inventory formats for nonroad mobile sources. The FF10 (Flat File 10) format is list directed (comma or semicolon delimeted) and the header of the file is used to indicate the nonroad mobile source data is within the file.

- **On-road mobile sources:** The Flat File 10 (FF10) format is list directed (comma or semicolon delimited) and contains activity inventory such as VMT, speed, and vehicle population data. This format requires VMT, SPEED, and VPOP inventory data.

- **Point sources:** SMOKE has formats for annual or average-day inventories, for day-specific inventories, and for hour-specific inventories. For annual or average-day inventories, the ORL and FF10 formats can be used for criteria, particulate, and toxics inventories. Finally, the CEM data format can be used for day-specific or hour-specific data : SMOKE uses the ORIS codes and boiler codes in the annual inventory files to match sources from the CEM data files.

- **Wildfire sources:** There are two approches available that you can provide wildfire data that are being treated as point sources to SMOKE using the ORL and FF10 point-source formats.
    - **Precomputed plume rise approach:** Certain fields must be left blank (such as stack parameters) because they do not apply to wildfire sources. When using wildfire data provided as point sources, you must also provide day-specific or hour-specific wildfire emissions and hour-specific precomputed plume rise using the FF10 day-specific and hour-specific formats.
    - **Internal plume rise calculation approach:** Requires two separate inventory files that are provided in a modified ORL format: (1) a list of fires with fire-specific characteristics including county/state/county, fire identification, location coordinate, fire name, SCC and others, as described in Section 8.2.8.3, "ORL FIRE Format" [380], and (2) a day-specific fire data including size of area burned, fuel loading, and star/end hour of fire (Section 8.2.6.2, "ORL FIRE Emission Format" [368]). Unlike the approach listed above, this approach internally estimates the plume rise using the size of the area burned and fuel loading, and it adjusts temporal profiles using the start and end hours of the fire. See detail at Section 4.4.17, "Plume Rise Calculation for Fires" [167]

2.3.4. Country, state, and county codes

SMOKE uses a 6-digit integer code to identify a country, state (or province), and county (or other region) for a particular source. Most U.S. inventories input to SMOKE have the 5-digit U.S. Federal Implementation Planning Standards (FIPS) state and county codes. All inventory input formats have been adapted to include a special header record with which you can specify the country, effectively allowing the inventories to be provided with the 6-digit code that SMOKE uses. The 6-digit system was designed for use in the United

States with states and counties, as well as Canada and Mexico, but it can be adapted for other uses. The format used by SMOKE for the codes is:

C SS YYY
Country State County
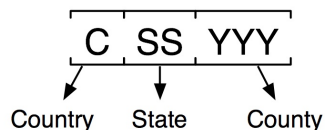
The SMOKE installation is set up to use U.S.-centered codes as defined in the `COSTCY` or the `GEO-CODE_LEVEL4` (if USE_EXP_GEOCODES Y) file, which contains the codes and their associated names and time zone settings. In this file, the U.S. country code is zero, which allows the U.S. country/state/county codes to be the same as the FIPS state/county codes that appear in U.S. inventories. See Section 8.10.2, "`COSTCY`: Country, state, and county names and data" [445] for more information on the `COSTCY` file format.

To change the meaning of the country, state, or county codes in SMOKE, the `COSTCY` or the `GEO-CODE_LEVEL4` (if USE_EXP_GEOCODES Y) file must be modified to use different names associated with each country, state, county or tribal number. All SMOKE input files must also use this new numbering scheme, including inventory files and cross-reference files.

Acceptable values in SMOKE for the country code are 0 through 9. Acceptable values of the state code are 1 through 99. Acceptable values of the county code are 1 through 999. No alphabetic codes are accepted, since SMOKE stores these values as integers.

2.3.5. Source Classification Codes

EPA uses Source Classification Codes (SCCs) and area and mobile source (AMS) codes to classify different types of anthropogenic emission activities. SCCs have 8 digits for point sources, while AMS codes have 10 digits, and sometimes include a leading "A" as an eleventh character. In SMOKE, we refer to both kinds of codes as "SCCs", and we ignore the leading "A" in the area and mobile codes. Additionally, SMOKE permits the nonpoint and point toxics inventories to use both 8-digit and 10-digit SCCs in the same inventory input file, because both 8- and 10-digit codes are contained in the nonpoint and point inventories in the 1999 NEI for HAPs. The maximum field width in SMOKE and its input files for SCCs is 20 characters as of SMOKE v4.0. The 8 or 10 digit SCC are still supported, but if the SCC is greater than 10 digits the SCC hierarchial approach will not be supported.

For SCC's of size 10 characters or less, the codes use a hierarchical system in which the definition of the code gets increasingly more specific as you move from left to right. (NOTE: if the SCC is greater than 10 characters in length the hierarchial system is not used). For SCC's of 10 characters or less, it is important to understand the hierarchy of the codes, because you can take advantage of the hierarchy in building cross-reference files for assigning emissions processing factors to inventory emission sources. In the diagrams below, level 1 is the least specific and level 4 is the most specific.

The code structure for the 8-digit point-source codes is:

A BB CCC DD
Level 1 Level 2 Level 3 Level 4
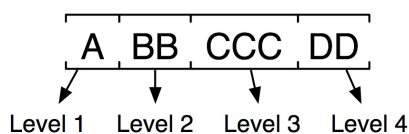
An example point-source activity and corresponding SCC can be taken directly from SMOKE's SCC description file (`SCCDESC`): "External Combustion Boilers; Electric Generation; Lignite; Spreader Stoker"

is represented by 10100306. Below we have mapped the levels of this description with the levels of the SCC:
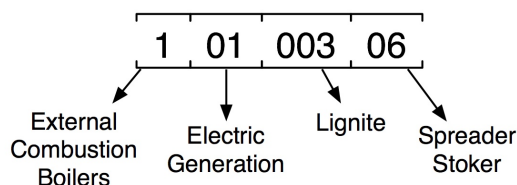
Similarly, the code structure for the 10-digit area- and mobile-source codes is:
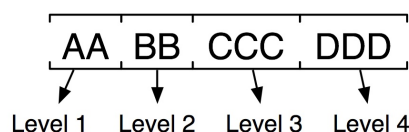
SMOKE treats SCCs as character strings, though in practice the values in the inventories and cross-reference files are usually numeric. In Section 2.4, "Cross-referencing and profiles" [17] on cross-references and profiles, we explain how these hierarchies are used by SMOKE and how you should use them in preparing SMOKE input files.

For on-road mobile sources, SCCs are treated somewhat differently than for other source categories. We explain more about this in Section 2.8.4.1, "Special approach for on-road mobile MOVES SCCs" [37].

2.3.6. Standard Industrial Classification codes

Although SIC codes are being replaced by NAICS codes in building emission inventories at EPA, SIC codes are still used in emissions processing. As of SMOKE v4.0, the SIC codes may be up to 20 characters in length, but for SIC codes greater than 4-digits, the hierarchial system is not used. For SICs of size 4-digits, a 2-level hierarchial system is recognized by SMOKE for application of growth, control, and chemical speciation factors. The two code levels are illustrated below.
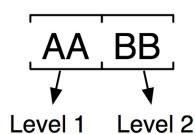
2.3.7. Geographical Code (GEOCODE_LEVEL[1-4]

Geographical codes may be specified to the user's desired level of detail using the GEOCODE_LEVEL[1-4] files. GEOCODE_LEVEL1 contains three character codes for each country in the inventory (CCC). GEOCODE_LEVEL2 contains six character codes for each state that the user would like to track in the inventory (CCCSSS). GEOCODE_LEVEL3 contains nine character codes for each county that the user would like to track in the inventory (CCCSSSYYY). GEOCODE_LEVEL4 contains twelve character codes for each tribal region that the user would like to track in the inventory (CCCSSSYYYTTT).

CCC SSS YYY TTT

Country State County Tribal
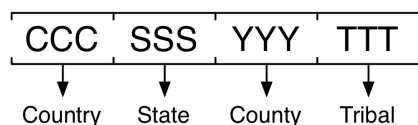
2.3.8. Maximum Achievable Control Technology codes

The following quote explaining MACT codes was taken from EPA to explain what MACT codes are and why they are used in some inventories and not others:

> To evaluate EPA's progress in reducing air toxic emissions through the Maximum Achievable Control Technology (MACT) standards and to identify sources that may be modeled as part of residual risk assessments, operations within facilities that are subject to MACT standards are identified in the NTI by 4-digit MACT codes. *[note that the term NTI (National Toxics Inventory) has since been replaced with NEI and that the codes are now 6 digits]*
>
> A MACT category is one for which emissions limitations have been or are being developed under section 112(d) of the Clean Air Act (National Emissions Standards for Hazardous Air Pollutants). EPA sets source category, technology based standards through its MACT program that sharply reduce emissions of HAPs. EPA's ATW web site includes information on the MACT source categories and the MACT program (www.epa.gov/ttn/atw/eparules.html). The tagging of data with MACT codes allows EPA to determine reductions attributable to the MACT program. The NTI associates MACT codes corresponding to MACT source categories with stationary major and [section-112] area source data. MACT codes are assigned at the process level or at the site level in the point source data, e.g., the MACT code for municipal waste combustors (MWCs) is assigned at the site level whereas the MACT code for petroleum refinery catalytic cracking is assigned at the process level. MACT codes are also assigned to source categories in the nonpoint source file.

In SMOKE, MACT codes are treated as 6-character strings, with no internal hierarchy associated with the number.

2.3.9. Source types: major and section-112 area sources

For point and nonpoint toxics inventories, each source can be labeled as "major" or "section-112 area" for input to SMOKE (the following paragraph explains how the term "area" can be applied to a point-source inventory). The Clean Air Act defines major sources as those stationary facilities that emit or have the potential to emit 10 tons per year or more of any one toxic air pollutant or 25 tons per year or more of any combination of toxic air pollutants. section-112 area sources include facilities that have air toxics emissions below the major source threshold as defined in section 112 of the Clean Air Act and thus emit less than 10 tons per year of a single toxics air pollutant or less than 25 tons per year combined of multiple toxics air pollutants. Another source type exists in principle for nonpoint sources: the "other" source type; an example of this source type is wildfires. However, these source types are not labeled differently from the section-112 area sources in the nonpoint toxics inventories, so the "other" source type has not been included in SMOKE to date.

A note about the confusing use of "area" terminology to describe point sources: The designation of sources in the point inventories as section-112 area sources has no relationship whatsoever to SMOKE's area

processing category. The point sources that are section-112 area sources are still processed by SMOKE as point sources using a lat-lon location and stack parameters.

In practice, all "major" sources should appear only in the point toxics inventory, but in some cases, "major" sources have shown up in the nonpoint inventory (specifically in inventory year 1996, in the July 2001 version of that inventory). Thus, the source type designation is provided in both the point and nonpoint toxics input formats.

The major and section-112 area designations are used when applying MACT-based control factors. These control factors are assigned based on a source's MACT code and may be applied to major sources only, to section-112 area sources only, or to both types of sources regardless of designation.

2.3.10. Source types: nonroad and onroad mobile sources

The nonroad and onroad mobile source type designations are used when applying MACT-based control factors. These control factors are assigned based on a source's MACT code.

2.4. Cross-referencing and profiles

The emission inventories described in Section 2.3, "Emission inventories" [6] can contain hundreds of thousands or even millions of sources. Collecting specific information for each source about its temporal allocation, chemical speciation, and spatial allocation is not practical. Therefore, a part of emissions processing involves assuming that many sources share the same factors for these major processing steps. For example, we apply monthly, day-of-week, and hourly temporal factors (called profiles) to convert from an annual emissions value to an hour-specific emissions value. A limited set of monthly, day-of-week, and hourly diurnal profiles are available from various studies, and these profiles each have their own unique profile number (also called profile code or profile ID). This limited set of profiles is assigned to the much more numerous inventory sources using an approach called cross-referencing, which is implemented using cross-reference files.

The cross-reference files assign the profiles based on source characteristics such as country, state, and county codes and/or SCCs, using the profile numbers to associate source characteristics with the profiles. While the profile numbers are unique in the profile files, they will appear many times in the cross-reference; this is how SMOKE is able to group the sources to treat them in the same manner. This approach is used for temporal allocation profiles, chemical speciation profiles and the spatial "profiles", which are called spatial (or gridding) surrogates.

The cross-reference tables are applied to the sources in a stepwise manner, such that the most specific entry available is always applied. For example, if a cross-reference entry were available that matched a source by state, county, and SCC, SMOKE would apply that entry instead of a different cross-reference entry that matched that source only by SCC. The hierarchy that describes how each cross-reference file is applied to the inventory is described for each program in Chapter 6, *SMOKE Core Programs* [211].

Figure 2.1, "Generic example of how cross-reference files and profiles work together" [18] provides a generic example of how cross-reference files and profile files work together. In the example, the profile to be used for most of North Carolina is profile ID 16. Durham and Orange counties, however, are assigned profile 15, which would be preferentially applied to all sources in Durham and Orange counties, instead of using the general North Carolina profile. South Carolina sources would be assigned profile 17.

Figure 2.1. Generic example of how cross-reference files and profiles work together

Cross-reference file

State	County	ID
NC	Durham	15
NC	Orange	15
NC	All	16
SC	All	17

Profiles file

ID	Factors
13	0.2, 0.3, 0.5
14	0.4, 0.4, 0.2
15	0.4, 0.3, 0.3
16	0.4, 0.5, 0.1
17	0.4, 0.4, 0.2
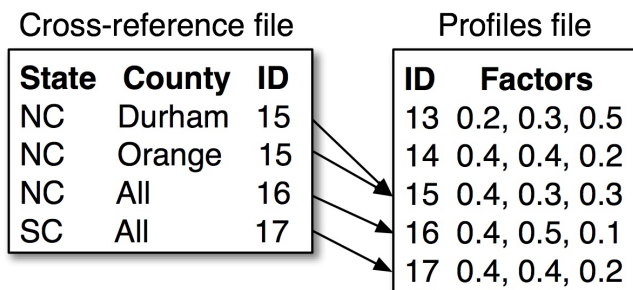
This example does not correspond to a particular processing step (i.e., temporal allocation, chemical speciation, or spatial allocation), but rather assigns generic "factors" from profiles 15, 16, and 17 based on the state and county information in the cross-reference file. (Note that we have used the state and county names in this example, whereas real cross-reference files would use the country, state, and county codes according to the file format of the actual cross-reference files.)

SMOKE handles cross-references and profile application in a very efficient manner. In reading a cross-reference file, SMOKE first sorts the cross-reference entries using the same sort criteria as are used for the inventory sources (e.g. by country/state/county code, then by SCC, then by remaining source characteristics if any). Next, the cross-reference entries are grouped according to the "level" of matching of each of the entries. For example, all entries that could match to the inventory using only state and county codes would be in one group, while entries that could match to the inventory using only SCCs would be in another group. Once the cross-reference entries are grouped, SMOKE processes each sources in the inventory, and attempts to find a matching entry in one of the cross-reference groups. The most specific groups are searched first, and when a match is found for a particular source, the other groups are not searched. This helps increase efficiency. In addition, because the cross-reference entries are sorted within each group, an efficient searching algorithm can be used for each individual search. When a match to one of the cross-reference groups has been found, SMOKE continues to the next source in the inventory until all sources have been processed.

Cross-references and profiles are used in the following SMOKE processing steps. These steps and their associated programs (listed in parentheses) will be described in the sections to come.

- Inventory import (**Smkinven**)
    - cross-references: NHAPEXCLUDE, VMTMIX, PSTK, ARTOPNT
    - profiles: none (factors are included in the cross-reference files when needed)
- Temporal allocation (**Temporal**)
    - cross-references: ATREF, MTREF, PTREF
    - profiles: ATPRO, MTPRO, PTPRO
- Chemical speciation (**Spcmat**)
    - cross-references: GSREF, GSCNV
    - profiles: GSPRO
- Spatial allocation (**Grdmat**)
    - cross-references: AGREF, MGREF
    - profiles: AGPRO, MGPRO (*** Note**)
- Growth and controls (**Cntlmat**)
    - cross-references: GCNTL
    - profiles: none (factors are included in the cross-reference files)
- Mobile-source speed assignment (**Movesmrg**)
    - cross-references: MCXREF, MFMREF
    - profiles: SPDPRO

The hierarchies that each SMOKE program uses to assign cross-reference entries to sources are provided in Chapter 6, *SMOKE Core Programs* [211], where the programs are described at length. The file contents and formats are described in more detail in Chapter 8, *SMOKE Input Files* [345].

Note: The use of the Environment variable AGPRO (Area spatial surrogate file)and MGPRO (Mobile spatial surrogate file) have been discontinued. Two new Environment variables have been introduced to SMOKE; SRGPRO_PATH (spatial surrogate profile file location) and SRGDESC (description file with the specific list of available surrogates located in SRGPRO_PATH) See Figure 6.4, "**Grdmat** input and output files" [230]. The surrogate files located in SRGPRO_PATH are refinements of the old [A|M]GPRO files. They are of the same format as the old files, however, there now may be one or more surrogate files. **Grdmat** now process each surrogate separately. On domains with large cell counts, this approach limits the memory usage at the expense of slightly longer run times.

2.5. Input and output file types

Before we describe more about the SMOKE processing, we first need to explain the types of files you will encounter in this documentation. SMOKE primarily uses two types of file formats: ASCII files and I/O API files. In addition, the output file format for the UAM-based air quality model is a Fortran binary file format. Chapter 8, *SMOKE Input Files* [345], Chapter 9, *SMOKE Intermediate Files* [461], and Chapter 10, *SMOKE Output Files* [483] describe all input, intermediate, and output files, including the file format for each one. Input files are files that are read by at least one core SMOKE program (listed in Chapter 6, *SMOKE Core Programs* [211]), but are not written by a core program. Intermediate files are files that are written by a core program and read by at least one other core program. Output files are files output by a SMOKE core program but not read by any of them; these files include reports, log files, and the model-ready files to be input to an air quality model. (Exception: one intermediate file [used by a core program] is also an output file [used by an AQM]: the STACK_GROUPS file, described in Section 9.3.2, "STACK_GROUPS" [464].) In this section, we further describe the ASCII and I/O API files, and then provide information about the two approaches for formatting the model-ready output files produced by SMOKE (the CMAQ/Models-3 approach and the UAM-based approach).

SMOKE's input files are primarily ASCII files, although a few I/O API files are used. The intermediate files in SMOKE are primarily I/O API files, although there are several important ACSII files as well. The output files from SMOKE are primarily I/O API files and Fortran binary files for model-ready emissions files, and ASCII files for reports and logs.

2.5.1. ASCII files

ASCII files are simply the text files with which most computer users are familiar. The ASCII files input by SMOKE come in two structures: *column-specific* and *list-directed*.

2.5.1.1. Column-specific ASCII files

In column-specific files, the fields in the files must appear in certain columns in the file. Each character on a line represents a single column. The lines below represent a column-specific ASCII data file:

```
TEST 1 2 3

Additional data
```

The letters TEST are in columns 1 through 4 of the file and the numbers 1, 2, and 3 are in columns 6, 8, and 10 respectively:

```
123456789012345
TEST 1 2 3

Additional data
```

2.5.1.2. List-directed ASCII files

In list-directed files, the exact positioning of the fields on a line is not important, but the order of the fields on that line is crucial. The fields must be delimited (separated) by special characters called delimiters; in SMOKE, valid delimiters are **spaces**, **commas**, or **semicolons**. If a particular field happens to contain any of these delimiters within it, then that field must be surrounded by single or double quotes in the input file.

2.5.2. I/O API files

I/O API files are read and written by the I/O API library used by SMOKE and other Models-3 programs. A library is a set of routines that have been created and compiled for use by multiple programs. The I/O API library, in turn, is built upon yet another library called the NetCDF library. For this reason, I/O API files are also referred to as I/O API NetCDF files. More information on both of these libraries is available at the I/O API web site [http://www.baronams.com/products/ioapi/]. Section 12.4, "Compiling SMOKE for UNIX" [496] contains instructions for obtaining the I/O API and NetCDF libraries.

The I/O API files cannot be viewed with a text editor because they are binary files. These binary files use less disk space than ASCII files containing the same data. They also allow much more efficient input and output of the data, and the I/O API library provides many quality assurance (QA) features useful for all input and output (I/O), including I/O for emissions processing.

The basic I/O API file has a limitation of 120 variables per file. To overcome this, SMOKE uses a wrapper called the FileSetAPI that creates and manages multiple I/O API files when more than 120 variables are needed in a single I/O API dataset in SMOKE. For example, if the SMOKE speciation matrix requires 140 pollutant-to-species variables, SMOKE will open by default two standard I/O API files: one with 120 variables and one with 20 variables. This resulting "file set" will be treated by other SMOKE programs as a single file, which enables processing of any number of pollutants and species in a single run, despite the I/O API variable limitation.

Some I/O API files can be viewed by the Visualization Environment for Rich Data Interpretation [http://www.verdi-tool.org] (VERDI). In SMOKE, any gridded output file from the **Smkmerge**, **Mrggrid**, or **Smk2emis** programs can be viewed by VERDI.

In some cases, it can be helpful to directly view the contents of the I/O API files in text form. This provides a quick way to check grid settings, time period, or species names in the model-ready output files. By viewing the text version of the model-ready output files produced by SMOKE, you can easily confirm that the correct species have been created or that the emission units are correct. To convert the I/O API files to text, one can use a combination of the NetCDF-provided **ncdump** utility and UNIX commands. The **ncdump** utility is created when you compile the NetCDF library, or you can download it from the NetCDF web site [http://www.unidata.ucar.edu/packages/netcdf/]. The command to convert the files to text format is:

`ncdump <infile> | cut -c1-80 > <outfile>`

Replace `<infile>` in the command above with your input I/O API file name, and `<outfile>` with your desired ACSII output file name. The output file contains all the applicable data stored in the I/O API file including grid information, time period, variable names, etc.

2.5.3. Model-ready files

SMOKE supports two major approaches for formatting its output files that are used as inputs to air quality models (i.e., model-ready files): the CMAQ/Models-3 approach and the UAM-based approach. The CMAQ/Models-3 approach is used for the CMAQ and MAQSIP models, and the UAM-based approach is used for the UAM models, REMSAD, and CAM$_X$.

The CMAQ/Models-3 approach uses one required 3-D I/O API file that contains the gridded, hourly, speciated, and vertically distributed emissions. In SMOKE, it is called the `EGTS3D_L` file. To create the 3-D model-ready emissions file, SMOKE computes plume rise for some or all point sources. For CMAQ, two additional optional files can be provided for plume-in-grid (PinG) processing. The first must contain locations and stack parameters for PinG sources and is called the `STACK_GROUPS` file. The second must contain the hourly, speciated emissions for the same PinG sources in a file called the `PINGTS_L` file. The MAQSIP model does not support PinG sources.

The UAM-based approach has two required files: (1) a 2-D emissions Fortran binary file with all sources other than point sources and all low-level point sources, and (2) an elevated-point-source Fortran binary file. The SMOKE program **Smk2emis** can create the 2-D emissions Fortran binary file (called the `UAM_EGTS` file) by converting a 2-D `EGTS_L` file from an I/O API format. To obtain the elevated-point-source Fortran binary file, the SMOKE program **Smkmerge** can create an ASCII elevated-point-source file, which can then be converted to the required binary format using the UAM preprocessor Ptsrce [http://www.remsad.com/ptsrce.htm].

2.6. Modeling parameters

Emissions modeling requires information about the subsequent air quality modeling that will be done. For example, to produce appropriate model-ready files using SMOKE, you must know which AQM will be used, the model grid and map projection, the episode dates, and the chemical mechanism to be used. In this manual, we refer to these settings collectively as "modeling parameters". In this section, we provide information on what these modeling parameters are and SMOKE's capabilities to support them.

SMOKE reads in the modeling parameters from both script settings (environment variables) and input files. In the subsections below, we provide the relevant settings and files that control the modeling parameters. More information about how to configure your scripts and files to change these parameters can be found in Section 4.4, "How to use SMOKE" [125]; how the settings affect the programs is described in Chapter 5, *SMOKE Utility Tools* [177] and Chapter 6, *SMOKE Core Programs* [211].

2.6.1. Map projections and model grids

A map projection is the mathematical representation of the spherical surface of the earth on a 2-D plane. SMOKE supports Lambert conformal, lat-lon, UTM, and polar stereographic map projections. There are many different settings that you may use to define your Lambert conformal, UTM, and polar stereographic projections, to make these projections match the one being used by your meteorology model and AQM. (Lat-lon is a fixed projection and cannot be changed.)

A model grid is a two-dimensional region overlaid on a map projection. It is defined by the starting *x-y* coordinates, the number of grid cells in each direction, and the physical size of the grid cells. Figure 2.2, "Example model grid" [22] shows an example of a model grid that includes most of the eastern U.S. This example has 81 grid cells in the *x*-direction, 75 grid cells in the *y*-direction, and each grid cell is 36 by 36 kilometers. Each set of 10 cells by 10 cells (counting from the starting coordinates) is enclosed in black grid lines.

Figure 2.2. Example model grid
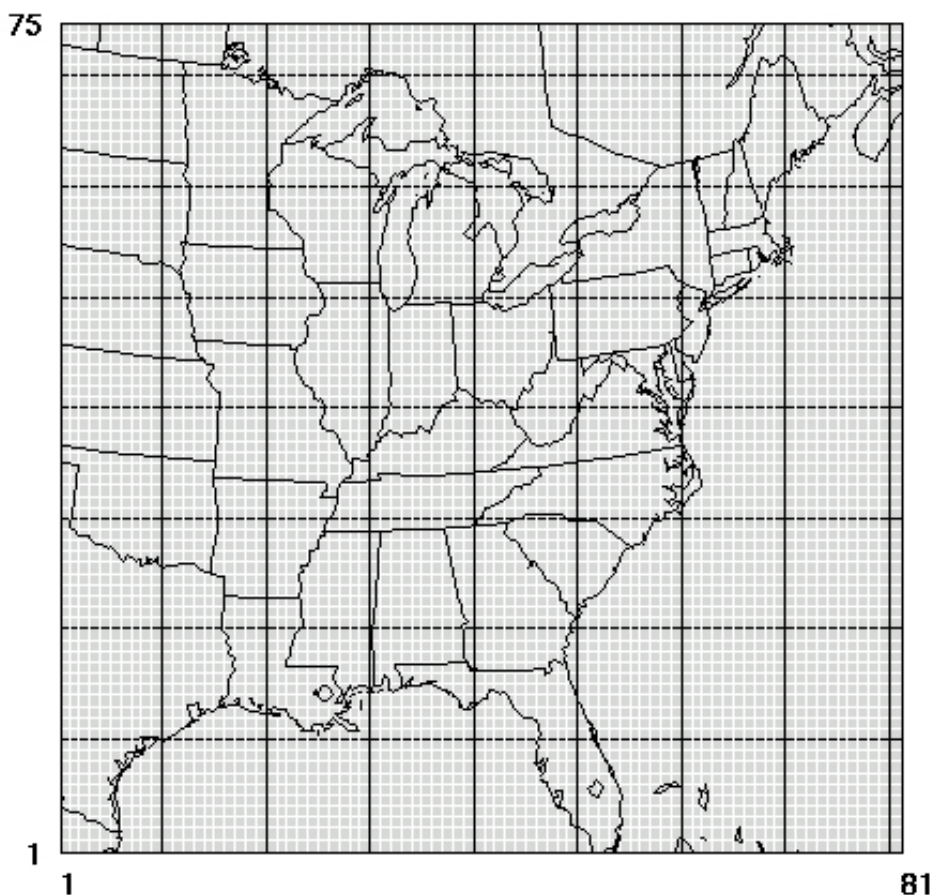
The model grid is set in SMOKE using the `IOAPI_GRIDNAME_1` setting to select a grid and map projection from among those defined in the `GRIDDESC` input file. The name of the grid set with the `IOAPI_GRID-NAME_1` setting must match a grid name in the `GRIDDESC` file to allow SMOKE to obtain the grid and map projection parameters from the `GRIDDESC` file.

2.6.2. Base year and past/future years

For any modeling effort, the emissions base year and future year are key modeling parameters needed for performing emissions processing. The base year is usually the year for which the air quality model is being run in order to compare modeling results with observed air quality data. Such comparisons allow modelers to tune the emissions data and air quality model, to ensure that the AQM is performing adequately during the modeling episode.

The base year is most often a year for which an emission inventory is available. This is usually the same year for which the meteorology model has been run to prepare input to SMOKE and an AQM and for which air quality observations are available. Of course there are exceptions to this principle, but generally that is how one establishes a base year.

Several different files and settings are used to set the base year in SMOKE, each of which should be consistent with each other for ideal results.

- The YEAR setting in the SMOKE Assigns file is the reference point used by the scripts to determine the base year and set the names of various year-specific input files.

- The episode and run settings (see Section 2.6.3, "Modeling episodes" [23]) determine the base year that will be used in the model-ready output files. This base year must match the YEAR setting so that the correct input files are used.

- The input emissions files should ideally contain data for the same base year, and the #YEAR header setting in those files should be consistent with the YEAR environment variable in the Assigns file. If the years in the annual inventory files are not consistent with each other, SMOKE will determine the year used by the most sources and set that as the base year. If day-specific or hour-specific data are used, all years in those files must be consistent with the base year of the annual emissions.

- The MOVES input data, if they are being used, should also be consistent with the base year. SMOKE is capable of running MOVES with inputs from a different year, but certain inputs may not be correct.

- Finally, the dates in the I/O API meteorology data from the Meteorology-Chemistry Interface Processor (MCIP) must be consistent with both the base year and the episode and run settings.

The future (or past) year is a chosen year in the future (or past) for which a modeler needs to run an air quality model; for example, to model the future effects of particular emission control strategies. To model a future year with SMOKE, you must have either an inventory that has been computed for a future year, or growth and control factors to project the base-year inventory to the future year. The settings and files that must be considered are as follows:

- The setting FYEAR is set in the run script and is used by the script to automatically assign the name of the **Cntlmat** input file GCNTL, which contains the growth factors. FYEAR must be set to the future year even if a future-year inventory is not being created because it has already been provided to you.

- If you already have a future-year inventory and so do not need to use SMOKE to project one from the base year inventory, then the emissions data year must match the future year, and the #YEAR header in the inventory file must match that year as well. In this case, the SMK_BASEYR_OVERRIDE setting must also be used to indicate what the base year is (which will be the same as the year of the meteorology data).

- The MOVES input data, if they are being used, must also include the correct settings for the future year of interest.

- The episode and run settings, meteorology files, and day- or hour-specific inventories should *not* match the future year, but rather should use the base-year episode dates.

2.6.3. Modeling episodes

The modeling episode is the total time period for which you will run SMOKE and your AQM. Unless the episode is just a few days long, users typically set up SMOKE to create emissions files of a shorter duration than their modeling episode, often creating one-day files for each day of their episode. Though SMOKE can create a single file for an entire episode, the file often becomes too large for some computers to handle (the limit for 32-bit operating systems is 2 GB files), so necessity rather than preference dictates that smaller files (usually one-day files) be created by SMOKE. We use the term "run period" to distinguish between these shorter durations and the full modeling episode; unless otherwise noted, we will assume that the run period is one day. For example, a typical SMOKE episode might cover July 1, 1996 through July 31, 1996. There will be 31 run periods (days) within this episode, the first starting on July 1, 1996 and the last starting on July 31, 1996.

In the SMOKE Assigns file, there are several settings that you need to change to cause SMOKE to create emissions for the episode of interest. Section 4.4, "How to use SMOKE" [125] provides more guidance on the particular form and approaches needed for using these settings.

- The episode start date (EPI_STDATE), episode start time (EPI_STTIME), episode duration in hours (EPI_RUNLEN), and the episode number of days (EPI_NDAY) all must be set to cover the modeling episode. Note that SMOKE can only be run for periods contained within a single calendar year. It cannot, for example, start in December of 1996 and run through January of 1997. Two separate episodes would need to be set up in this case, with the first ending on December 31, 1996, and the second starting on January 1, 1997.

- The start date of the first run period needs to be set using the G_STDATE and ESDATE settings. The G_STDATE is the year and Julian day setting used by the SMOKE programs; in our example above, G_STDATE would be set to 1996183, since July 1 is the 183rd day of 1996. The ESDATE is the Gregorian date used in naming the SMOKE intermediate and output files; for our example, ESDATE would be 19960701. The SMOKE scripts will use the EPI_NDAY setting to automatically loop through the number of run periods in the episode, starting with the first G_STDATE value in the Assigns file. The G_STDATE and ESDATE settings are changed for each run period.

- The run period start time (G_STTIME) and duration (G_RUNLEN) must also be set to indicate the start time and length of each run period. Both values are provided as a number of hours, using a HHMMSS (hours, minutes, seconds) format.

  The run period duration (G_RUNLEN) is usually not the same as the episode duration (EPI_RUNLEN). For example, if the episode length is 30 days (720 hours), the run period duration setting could be just 1 day (25 hours), 2 days (49 hours), or three days (73 hours) (the reason for the extra hour in each case is explained below). In the first case, SMOKE would create thirty 25-hourfiles; in the second case, fifteen 49-hour files; and in the third case, SMOKE would create ten 73-hour files.

- The NDAYS, MSDATE, and MDAYS settings are used for naming files. The NDAYS setting should be set to the number of days in each run period, and is used by default for naming time-based files. The NDAYS setting is also used along with the EPI_NDAY setting to loop through the run periods in the episode. The MSDATE and MDAYS settings can be used for naming the meteorology input files, but are not being used by the default Assigns file provided with SMOKE.

There are a few key things to remember when you are verifying that you have the correct episode settings:

- SMOKE cannot process emissions over a calendar-year break. Thus, the longest run that can be done is for 365 days, with the episode start date being January 1. If a modeling episode spans multiple years, then a different Assigns file, script, and sets of input files must be created for each year.

- The AQMs supported by SMOKE always need one extra hour in each emissions input file due to how they calculate boundary conditions. Therefore, if you are inputting emissions to run a 24-hour period, the G_RUNLEN setting should be 250000 for 25 hours.

- The CMAQ, REMSAD, and CAM$_X$ models can accept emissions files for multiple days, but the UAM must have 25-hour files only. As stated earlier, however, all of these models are often run using 25-hour files, with one file for each day of the episode.

- All times are associated with a time zone, including the episode and run period start time settings. These settings must be consistent with the time zone of the meteorology files. If the meteorology data were created using MM5, the time zone is most likely Greenwich Mean Time (GMT); therefore, the EPI_STDATE, EPI_STTIME, G_STDATE, and G_STTIME settings would have to be provided in that same time zone. Whatever time zone is inherent in the meteorology files and these date settings

will also be the time zone of the dates and times in the output emissions files from SMOKE. This ensures that the dates and times of the emissions and meteorology files are consistent for input to the AQM.

2.6.4. Chemical mechanisms

SMOKE can accommodate a variety of chemical mechanisms for the models it supports. From the emissions processing perspective, the chemical mechanism is the mapping of the pollutants provided in the emissions inventory to the species needed by the AQM of interest. For example, the input files for five chemical mechanisms for the CMAQ model are available for download from the EPA; these mechanisms are Carbon Bond 4 (CB4), CB4 with particulates, Regional Acid Deposition Model, 2 (RADM2), RADM2 with particulates, and a research version of CB4 with toxics.

In Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150], we provide the settings needed in the Assigns file to use a different chemical mechanisms with SMOKE. SMOKE is not constrained to the files available for download. If you need to process other data (e.g., a tracer species) with SMOKE, they can be added to several input files, including the chemical mechanism file, to be output to the AQM. Some additions to chemical mechanisms are easier than others, and we explain how to determine whether you can create the files you need for your situation. We also give instructions on how to add species to the chemical mechanism files and how to make sure that the inventory pollutants are mapped to the correct chemical species.

SMOKE users must know what chemical mechanism will be used in the AQM for which the SMOKE output emissions are intended. Once that has been determined, the following files must be configured to be consistent with the inventory being used and the chemical mechanism: the inventory table (`INVTABLE`), speciation profiles (`GSPRO`), speciation cross-reference (`GSREF`), and the mobile processes file (`MEPROC`) when creating on-road mobile emissions with MOVES through SMOKE.

2.6.5. Layer structures

SMOKE needs information on layer structures for processing elevated point sources' plume rise in the **Laypoint** program and creating the ASCII elevated-point-source file (`ELEVTS_L` or `ELEVTS_S`) with the **Smkmerge** program. The way SMOKE obtains the layer information differs depending on whether you are creating emissions using a CMAQ-based or UAM-based approach (see Section 2.5.3, "Model-ready files" [21]). For the CMAQ-based approach, SMOKE determines the layer structure from the structure included in the header of the `GRID_CRO_3D` meteorology file. For the UAM-based approach, SMOKE does not really need to know the layer structure, except to output it to the ASCII elevated-point-source file. In this case, there are many settings obtained by **Smkmerge** from environment variable names starting with `UAM_`. These appear in the REMSAD run scripts provided with SMOKE and must be set to be consistent with the layer information expected by UAM, CAM$_X$, or REMSAD.

2.7. Sparse matrix approach to emissions modeling

The paradigm for atmospheric emissions models prior to SMOKE was a network of pipes and filters. This means that at any given stage in the processing, an emissions file includes self-contained records describing each source and *all* of the attributes acquired from previous processing stages. Each processing stage acts as a filter that inputs a stream of these fully-defined records, combines it with data from one or more support files, and produces a new stream of these records. Redundant data are passed down the pipe at the cost of extra I/O, storage, data processing, and program complexity. Using this method, all processing is performed one record at a time, without necessarily a structure or order to the records.

This old paradigm came about as a way to avoid repeatedly searching through data files for needed information, which would be very inefficient. It is admirably suited to older computer architectures with very small available memories and tape-only storage, but is not suitable for current desktop machines or high-performance computers. SMOKE developers demonstrated this when the Emissions Preprocessor System (EPS) 2.0 was run on a Cray Y-MP. It ran four times slower on the Cray machine (a much faster computer) than on a desktop 150 MHz DEC Alphastation 3000/300. This paradigm also fostered a serial approach to the emissions processing steps, as shown in Figure 2.3, "Serial approach to emissions processing" [26].

Figure 2.3. Serial approach to emissions processing
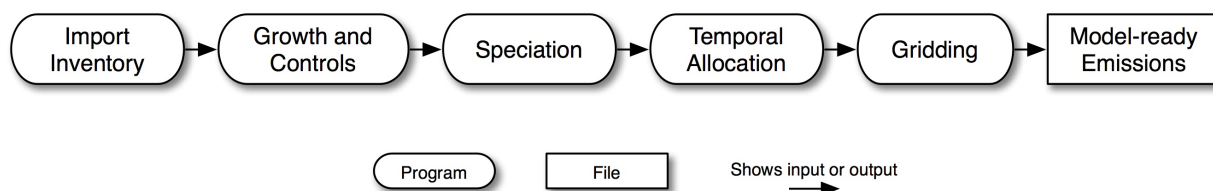
The new paradigm implemented in SMOKE came about from analyses indicating that emissions computations should be quite adaptable to high-performance computing if the paradigm were appropriately changed. For each SMOKE processing category (i.e., area, biogenic, mobile, and point sources), the following tasks are performed:

- read emissions inventory data files

- optionally grow emissions from the base year to the (future or past) modeled year (except biogenic sources)

- transform inventory species into chemical mechanism species defined by an AQM

- optionally apply emissions controls (except for biogenic sources)

- model the temporal distribution of the emissions, including any meteorology effects

- model the spatial distribution of the emissions;

- merge the various source categories of emissions to form input files for the AQM

- at every step of the processing, perform quality assurance on the input data and the results

Each processing category has its particular complexities and deviations from the above list; these are described in Section 2.8, "Area, biogenic, mobile, and point processing summaries" [29]. For all categories, however, most of the needed processing steps are *factor-based*; they are linear operations that can be represented as multiplication by matrices. Further, some of the matrices are *sparse* matrices (i.e., most of their entries are zeros).

SMOKE is designed to take advantage of these facts by formulating emissions modeling in terms of sparse matrix operations, which can be performed by optimized sparse matrix libraries. Specifically, the inventory emissions are arranged as a vector of emissions sorted in a particular order, with associated vectors that include characteristics about the sources such as the state/county and SCCs. SMOKE then creates matrices that apply the control, gridding, and speciation factors to the vector of emissions. In many cases, these matrices are independent from one another, and can therefore be generated in parallel and applied to the inventory in a final "merge" step, which combines the inventory emissions vector (now an hourly inventory file) with the control, speciation, and gridding matrices to create model-ready emissions. Figure 2.4, "Parallel approach to emissions processing" [27] shows how the matrix approach allows for a more parallel approach to emissions processing, in which fewer steps depend on other needed steps.

Note that in Figure 2.4, "Parallel approach to emissions processing" [27], temporal allocation outputs hourly emissions instead of a temporal matrix. This is because of some peculiarities with temporal modeling for point sources, which can use hourly emissions as input data. To be able to overwrite the inventory emissions with these hourly emissions, the temporal allocation step must output the emissions data. The matrix approach is used internally in the temporal allocation step.

The growth and controls steps shown in Figure 2.4, "Parallel approach to emissions processing" [27] are optional. If the inventory is not grown to a future or past year, then the temporal allocation step uses the original inventory vectors to calculate the hourly emissions.

Figure 2.4. Parallel approach to emissions processing
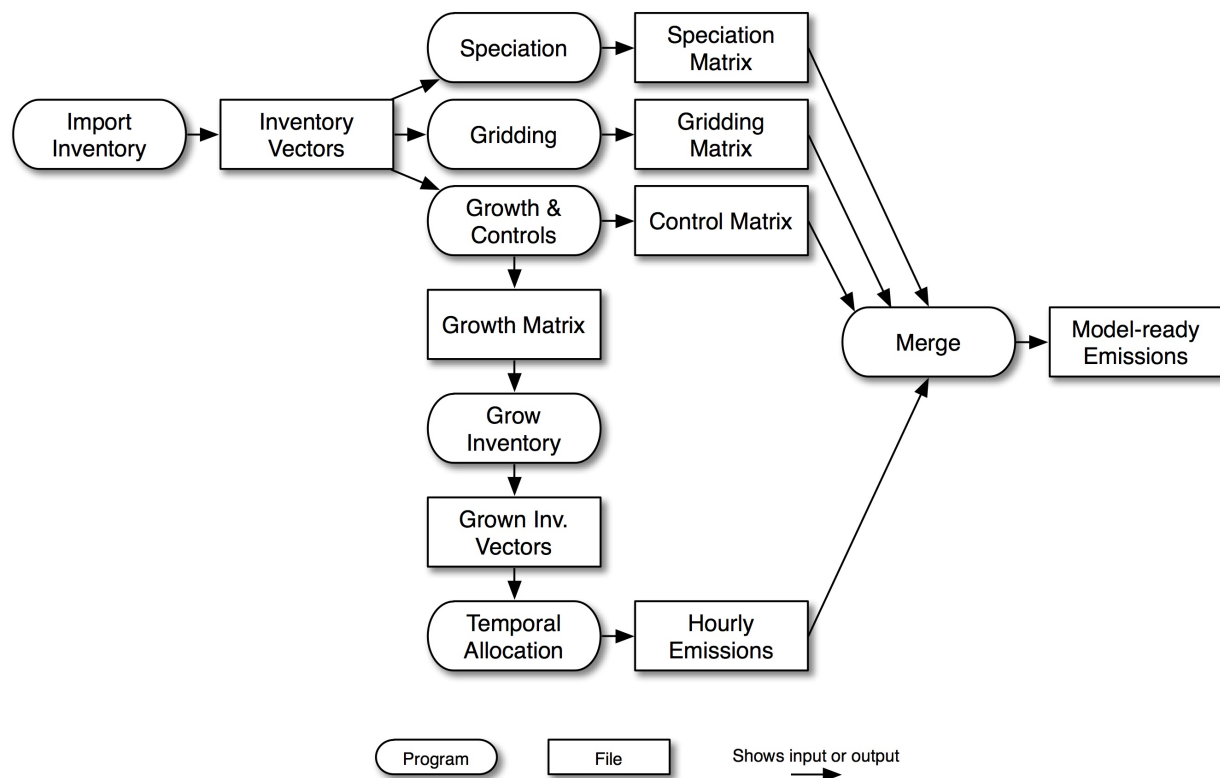
Several benefits can be realized from this more parallel approach. For example, given a single emissions inventory, temporal modeling is performed only once per inventory and episode (though in practice, this step is often performed once per episode day). Also, gridding matrices typically need only be calculated once per inventory and model grid definition, without having to reprocess other steps. As shown in Figure 2.5, "Processing steps for running an additional grid in SMOKE" [28], SMOKE usually needs to rerun only the gridding and merge steps to process a different grid for the same inventory. The merge step in the figure will read the previously created results from the temporal allocation, chemical speciation, and control processing steps.

Figure 2.5. Processing steps for running an additional grid in SMOKE
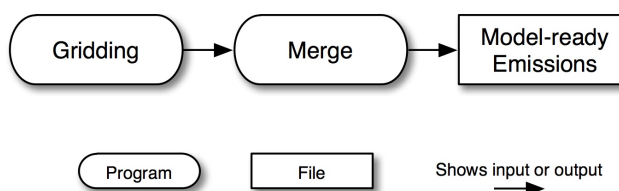
In addition, speciation matrices need only be calculated once per inventory and chemical mechanism. Similar to the gridding example, Figure 2.6, "Processing steps for running an additional chemical mechanism in SMOKE" [28] shows the SMOKE steps that generally need to be rerun for running an additional chemical mechanism.

Figure 2.6. Processing steps for running an additional chemical mechanism in SMOKE
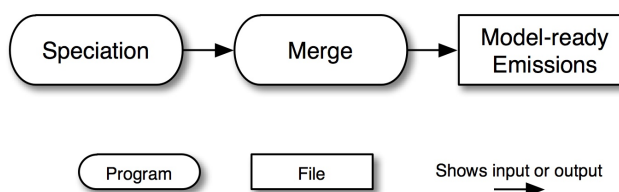
A final example of how this approach is beneficial is processing with a control strategy. Because of SMOKE's parallel processing, changing a control strategy requires only the control and merge steps to be processed again (Figure 2.7, "Processing steps for running a control scenario in SMOKE" [28]). In serial processing, on the other hand, the growth and controls step occurs as the second processing step, which requires that all downstream steps be redone. In Figure 2.7, "Processing steps for running a control scenario in SMOKE" [28], the speciation, temporal allocation, and gridding steps have already been run, and can be fed to the merge step without being altered or regenerated.

Figure 2.7. Processing steps for running a control scenario in SMOKE
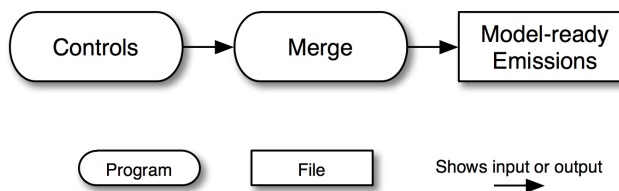
Although SMOKE processing generally follows the structure shown in Figure 2.4, "Parallel approach to emissions processing" [27], there are some exceptions. In the list below, we summarize these exceptions and provide references to the sections of this chapter where these exceptions are explained and shown through diagrams. These exceptions are also described in more detail in Section 2.8.2, "Area-source processing" [30], Section 2.8.3, "Biogenic-source processing" [33], Section 2.8.4, "Mobile-source processing using MOVES" [34], and Section 2.8.5, "Point-source processing" [47].

• **On-road mobile processing with MOVES:** One way of processing on-road mobile-source emissions is to have SMOKE run the MOVES model based on hourly, gridded meteorology data. To run a different grid or control strategy using this approach, users usually need to run a number of additional processing steps that we have not yet discussed. These differences from the standard processing approach are described in Section 2.8.4, "Mobile-source processing using MOVES" [34].

- **Biogenics processing:** Biogenics processing uses different processors than those for anthropogenic sources. The emissions from biogenic sources are based on land use data and meteorology data instead of on actual emission inventories. For more information, please see Section 2.8.3, "Biogenic-source processing" [33].

- **Toxics processing for different chemical speciation mechanisms:** Toxics processing may require some special processing steps during import of the inventory data when integrating the criteria and toxics inventories. This step depends on which chemical speciation approach is going to be used. Therefore, when changing the toxics speciation mechanism, it is sometimes necessary to rerun the data import step. See Section 2.9.5, "Combine toxics and criteria inventories" [55] for more information.

- **Point-source processing for CMAQ or MAQSIP versus UAM, REMSAD, or CAM$_X$:** Point-source processing for CMAQ or MAQSIP uses some different programs than processing for UAM, REMSAD, or CAM$_X$. In some cases, it may be necessary to rerun several programs in order to run for one model rather than another. Further details on this additional processing can be found in Section 2.8.5, "Point-source processing" [47].

- **Adding hour-specific or day-specific point-source data:** If you want to add hour-specific or day-specific point-source data after a point source run has already been performed, several processing steps must be rerun. Further details on this additional processing can be found in Section 2.8.5, "Point-source processing" [47].

2.8. Area, biogenic, mobile, and point processing summaries

2.8.1. Summary of SMOKE processing categories

Each SMOKE processing category is defined by its source *characteristics*, which correspond to the identifiers used in creating the emission inventory (e.g., state/county FIPS code and SCC). The processing categories also have source *attributes*, which are the other useful data in the emission inventories that SMOKE uses; examples are point-source flue gas exit height and temperature. Source characteristics *define* the sources as area, biogenic, mobile, or point sources and also distinguish one source in the inventory from another. Source attributes are additional data about the source that do not contribute to the source's uniqueness in SMOKE. We have previously described in Section 2.3.1, "Inventory data types" [7] the data types and the data attributes that are contained in the inventories that SMOKE uses. In the subsections below, we summarize the source characteristics of area, biogenics, mobile, and point sources. Please refer to Table 2.1, "Inventory source categories and SMOKE processing capabilities and categories" [9] for more information about how SMOKE processing categories map to the inventory source categories.

In SMOKE, each processing category is defined by source characteristics as follows:

- **Area sources** are defined by (1) country, state, and county codes, (2) SCCs, and (3) optionally, grid cell.

- **Biogenic sources** are defined differently depending on the type of processing you are using. They can be defined either by (1) country, state, and county codes and (2) land use code, or by (1) grid cell and (2) land use code.

- **Mobile sources** are defined by (1) country, state, and county codes, (2) SCCs, and (3) optionally link codes.

- **Point sources** are defined by (1) country, state, and county codes, (2) plant/facility codes, and (3) characteristics 1 through 5, one of which must be the SCC.

2.8.2. Area-source processing

In SMOKE, there are two major processing routes that you can take for area sources: the typical route and the pregridded data route. (Recall that by "area sources" in SMOKE we mean stationary area/nonpoint sources and nonroad mobile sources.)

2.8.2.1. Typical route

The typical route involves processing data identified by country/state/county codes and SCCs. The processing steps vary depending on whether you are doing base-case processing or future- or past-year processing. The steps for base-year processing are shown in Figure 2.8, "Base case area-source processing steps" [30]. In Figure 2.4, "Parallel approach to emissions processing" [27], we also included the major intermediate vectors and matrices; please refer to that diagram for those details. The inventory import step reads the raw emissions data, screens them, processes them, and converts the raw data to the SMOKE intermediate inventory file (inventory vectors in Figure 2.4, "Parallel approach to emissions processing" [27]). The emissions in the inventory file are subdivided to hourly emissions during temporal allocation; assigned chemical speciation factors during speciation, and assigned spatial allocation factors during gridding. The merge step combines the hourly emissions, speciation matrix, and gridding matrix to create model-ready emissions.

Figure 2.8. Base case area-source processing steps
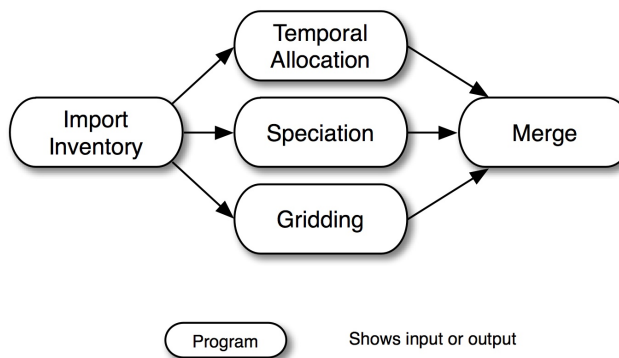
In Figure 2.9, "Future- or past-year growth and optional control area-processing steps" [31], we show the area-source processing steps for future- or past-year processing. This processing is similar to the base-year processing flow, except the growth and controls step is added to create the growth matrix and optionally one or more control matrices. The grow inventory step applies the growth matrix to convert the base-year inventory to a future or past year. Also, the control matrix can optionally be used in the merge step to apply control factors to the future- or past-year emissions. The steps shown with dotted lines represent steps that can be reused from the base-year processing because they do not depend on any of the new steps.

Figure 2.9. Future- or past-year growth and optional control area-processing steps
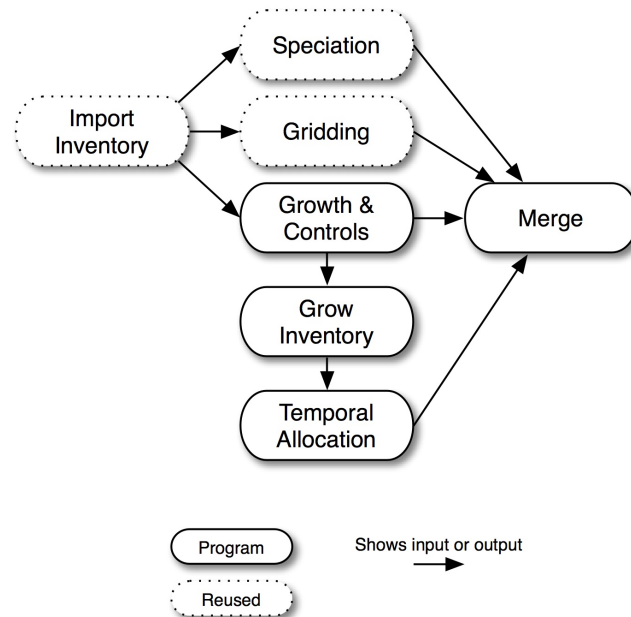
Finally, inventory controls as well as growth can be applied at the front end of processing if such a scheme is needed (Figure 2.10, "Alternative future- or past-year growth and control area-processing steps" [32]). This method permits up to 80 growth and/or control matrices to be applied to an inventory, whereas the method shown in Figure 2.9, "Future- or past-year growth and optional control area-processing steps" [31] allows only one control matrix in the merge step, although any number of growth matrices on the front end. The processing scheme shown in Figure 2.10, "Alternative future- or past-year growth and control area-processing steps" [32] can therefore be useful when mixing and matching many control strategies for simulations.

Figure 2.10. Alternative future- or past-year growth and control area-processing steps
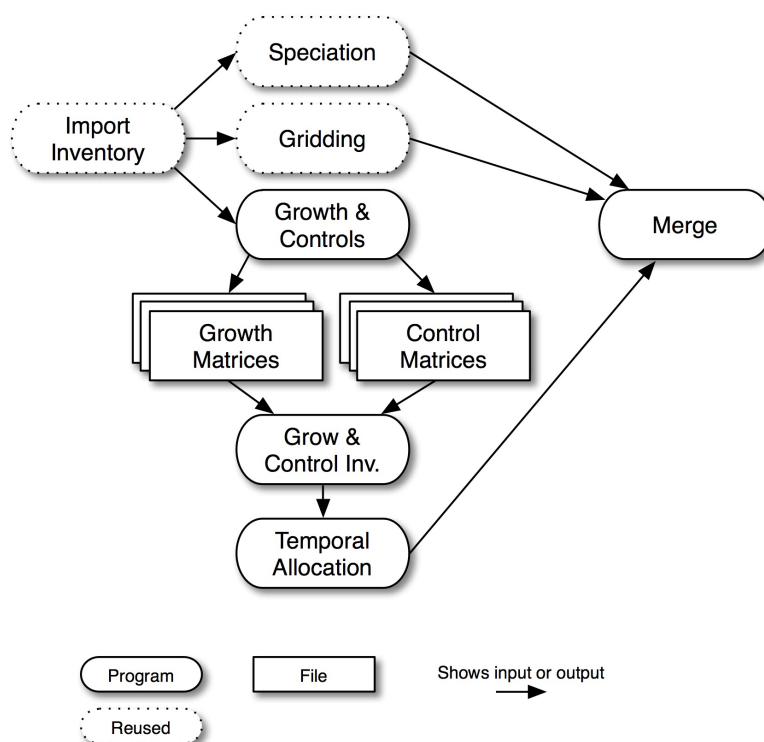
In sections later in this chapter, we describe the SMOKE programs that are needed for each of these processing steps and additional details about what activities are accomplished during each step. These sections are:

- Section 2.9, "Inventory import" [53]
- Section 2.10, "Temporal processing" [63]
- Section 2.11, "Chemical speciation processing" [70]
- Section 2.12, "Spatial processing" [73]
- Section 2.13, "Growth processing" [77]
- Section 2.14, "Control processing" [80]
- Section 2.18, "Creating model-ready emissions" [92]
- Section 2.20, "Quality assurance" [94]

2.8.2.2.1. Pregridded data route for same modeling domain

The second processing approach for area sources involves using pregridded data. As indicated in Section 2.8.1, "Summary of SMOKE processing categories" [29], area sources can be specified by grid cell instead of by country/state/county code and SCC. This optional approach to modeling area sources requires the inventory emissions data to be gridded prior to inventory import. The gridded area sources do *not* have country/state/county codes or SCCs, and can be provided via an I/O API time-independent gridded data file. The flow diagrams that describe this type of processing are identical to those in Figure 2.8, "Base case area-source processing steps" [30], Figure 2.9, "Future- or past-year growth and optional control area-processing steps" [31], and Figure 2.10, "Alternative future- or past-year growth and control area-processing steps" [32]. Although the gridding step is quite trivial when the grid cell numbers are already specified, the gridding step must still be run to create a gridding matrix required for the merge step.

The disadvantage of using pregridded emissions for area-source processing is that there are no country/state/county codes and SCCs to use in the cross-referencing of any processing step. Therefore, temporal profiles, speciation profiles, growth factors, and control factors must be applied uniformly across the model grid by pollutant.

2.8.2.2.2. Pregridded data route converting to a different modeling domain (eg EDGAR to CMAQ ready emissions file)

The sequence for processing global emissions data for hemispheric CMAQ involves projecting the data from geographic (latlon) to polar stereographic coordinates, converting the inventory species to the terms required by the CMAQ chemical mechanism, and extrapolating the annual emissions to hourly estimates.

2.8.2.3. Day-specific and hour-specific emissions

Emissions from area sources are sometimes available as day- or hour-specific values. **Smkinven** can import the day- and hour-specific data, and it can also convert the hour-specific data to hour-specific temporal profiles. When these data are available, the **Temporal** program overrides the annual or daily emissions with the most specific data available. If day-specific data are available, **Temporal** uses them to overwrite the annual or average-day emissions during the time periods that these data are available. If hour-specific data are available, **Temporal** uses them to overwrite the annual, average-day emissions, or day-specific emissions data.

2.8.3. Biogenic-source processing

SMOKE biogenic emissions modeling can be accomplished with the Biogenic Emissions Inventory System, version 3 (BEIS3) approach using the processing scheme (Figure 2.11, "Biogenic-source processing steps and intermediate files" [33]). The raw land use inventory data are imported and output as normalized emissions. Meteorology adjustments are then applied to the normalized emissions to create hourly model-ready emissions estimates.

Figure 2.11. Biogenic-source processing steps and intermediate files
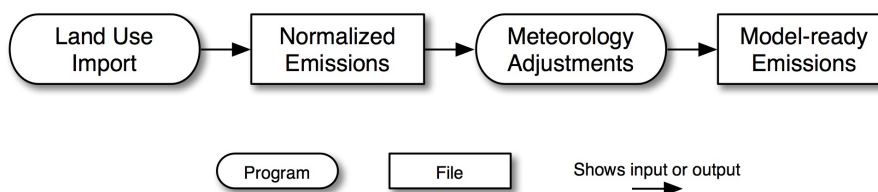
The land use import can start with gridded BELD3 or BELD4 land use data and uses BEIS3 summer and winter emission factors. In Section 2.17, "Biogenic processing" [91], we provide additional details about the SMOKE programs used for BEIS3 processing and its capabilities.

A variation can be run on the processing steps shown in Figure 2.11, "Biogenic-source processing steps and intermediate files" [33] (see Figure 2.12, "Biogenic-source processing steps and intermediate files using both winter and summer emission factors" [34]). In this variation, some grid cells use summer emission factors and some use winter emission factors. This is useful during the changes of seasons. Based on guidance from EPA, the summer emissions factors should be used for time periods after the last frost of the spring until the first frost of the fall, and winter emission factors should be used at other times of the year. To make such assignments by grid cell, the SMOKE utility **Metscan** analyzes the meteorology data for the entire year (or the period of interest) to establish which days each grid cell should use winter and summer emission factors. **Metscan** creates a winter/summer switch file that indicates the appropriate season for each grid cell for each day. More information on **Metscan** is available in Section 5.3.10,

"**Metscan**" [203]. The results of the meteorology analysis can then be used in the Figure 2.12, "Biogenic-source processing steps and intermediate files using both winter and summer emission factors" [34] processing approach, in which both the summer and winter normalized emissions are provided to the meteorology adjustments step, along with the winter/summer switch file. The resulting model-ready emissions data have used the winter emission factors for all grid cells of the domain that have experienced the first freeze date of the year but not the last (within a calendar year, this is the time periods January through March and November through December in many regions), and the summer emission factors for all grid cells between the last and first freeze dates.

Figure 2.12. Biogenic-source processing steps and intermediate files using both winter and summer emission factors
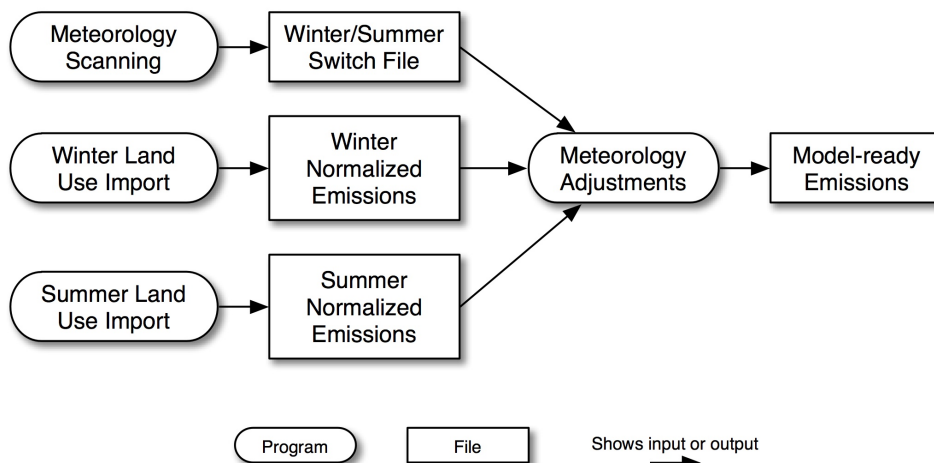
In Section 2.17, "Biogenic processing" [91] we describe the SMOKE programs that are needed for each of these processing types for BEIS3 processing, and additional details about what activities are accomplished during each step.

2.8.4. Mobile-source processing using MOVES

SMOKE provides two ways of processing mobile sources using MOVES. (Recall that by "mobile sources" in SMOKE we mean on-road mobile sources.) The first approach is to compute mobile emissions values prior to running SMOKE and provide them to SMOKE as input; we call this the precomputed-emissions approach. The second approach is to provide SMOKE with VMT data, Vehicle population (VPOP) data, meteorology data, and MOVES outputs, and have SMOKE compute the mobile emissions based on these data; this is called the MOVES approach. These approaches are not mutually exclusive, so it is possible to provide both precomputed emissions and VMT and VPOP data to SMOKE and have the system compute only some of the emissions using MOVES outputs. Both processing approaches can produce criteria, particulate, and toxics emissions results.

The precomputed-emissions approach is quite similar to the processing method for area sources. In fact, Figure 2.8, "Base case area-source processing steps" [30], Figure 2.9, "Future- or past-year growth and optional control area-processing steps" [31], and Figure 2.10, "Alternative future- or past-year growth and control area-processing steps" [32] from Section 2.8.2, "Area-source processing" [30] show exactly the processing steps needed for processing mobile sources using SMOKE and the precomputed-emissions approach. As in base-case processing for area sources, emissions in the inventory file are subdivided to hourly emissions during temporal allocation, assigned chemical speciation factors during speciation, and assigned spatial allocation factors during gridding. The merge step combines the hourly emissions, speciation matrix, and gridding matrix to create model-ready emissions. For future- or past-year processing, the growth

and controls step is added to create the growth and control matrices, while the grow inventory step converts the inventory from the base year to a future or past year. The control matrix can be optionally used in the merge step to apply control factors to the future- or past-year emissions. Note that, unlike the VMT approach, in the precomputed-emissions approach SMOKE will not model the variations in emissions caused by temperature, humidity, or other meteorological settings.

The MOVES approach is much different from the precomputed-emissions approach. Figure 2.13, "MOVES mobile RatePerDistance processing steps" [36] and Figure 2.14, "MOVES mobile RatePerVehicle and RatPereProfile (off-network) processing steps" [36] summarize the MOVES approach. First, county total activity inventory VMT data by road class and vehicle type or county total activity inventory VPOP by vehicle type are input to SMOKE. The chemical speciation step computes the chemical speciation factors for each county, road class, vehicle type, emissions process (e.g., exhaust start, exhaust running, evaporative processes, extended idle, and crankcase), and pollutant and stores the necessary factors for this transformation. The gridding step allocates the sources to grid cells and uses spatial surrogates to allocate county-total emissions to grid cells, storing the emission rates needed for these allocations based on hourly gridded ambient temperature meteorology.

The approach for running MOVES for SMOKE relies on the concept of reference counties and fuel months. The concept of reference county refers to running MOVES for a single county, which is the reference county, to represent itself and other counties that share the same MOVES input parameters and thus have the same emission rates for any given speed, temperature and humidity. A reference fuel month similarly refers to a reference fuel month's MOVES run that contains the temperatures that occur in neighboring months as well as the representative month. The mapping of calendar months to a representative month should be assigned on the basis of shared fuel parameters, because it is the interaction of fuel and temperature that is important. For example, an average-hourly temperature of 70°F may occur in some hour of any day in each of four months: May, June, July and August. If those four months share the same fuel properties (i.e. summer fuel) then an emission factor will be determined for just the representative month, reducing by a factor of four the number of calculations that MOVES needs to perform.

Unlike MOBILE6, MOVES differentiates between on-roadway emission processes and off-network emission processes. Figure 2.13, "MOVES mobile RatePerDistance processing steps" [36] summarizes the approach used by MOVES for on-roadway mobile sources. The on-roadway emission process includes county-total VMT and average speed inventory as input. The off-network emission processes use the county-total vehicle population by vehicle type as input. Figure 2.14, "MOVES mobile RatePerVehicle and RatPereProfile (off-network) processing steps" [36] summarizes the approach used by MOVES for off-network mobile sources. Both on-roadway and off-network emission processes do require real gridded meteorology data from MCIP files to estimate temperature-dependent emission rates.

Figure 2.13. MOVES mobile RatePerDistance processing steps
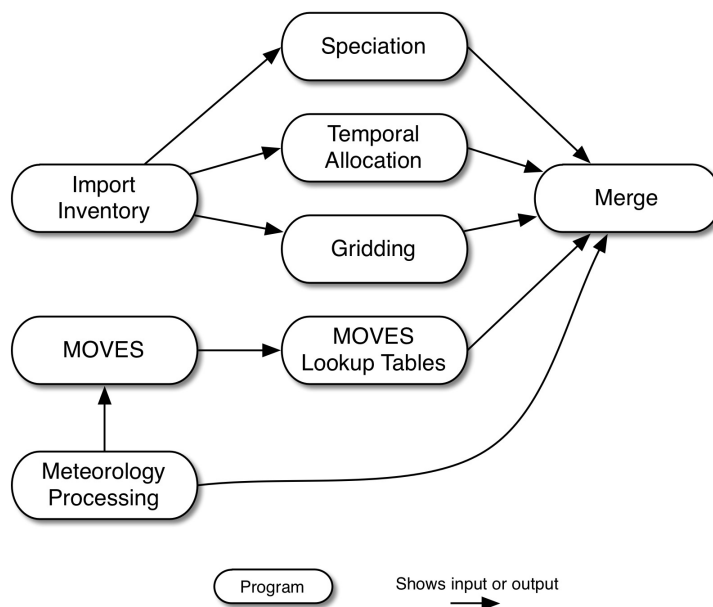
Figure 2.14. MOVES mobile RatePerVehicle and RatPereProfile (off-network) processing steps
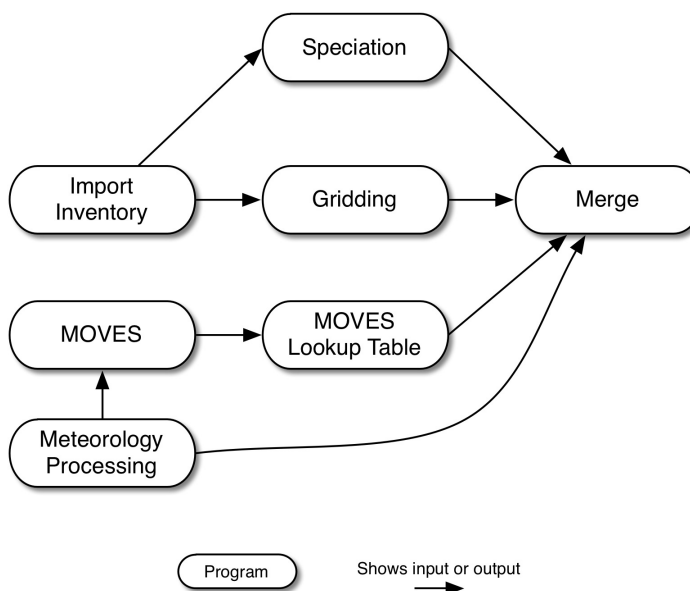
In sections later in this chapter, we describe the SMOKE programs that are needed for each of the processing steps just described for MOVES processed mobile sources, and additional details about what activities are accomplished during each step. These sections are:

- Section 2.9, "Inventory import" [53]
- Section 2.10, "Temporal processing" [63]
- Section 2.11, "Chemical speciation processing" [70]
- Section 2.12, "Spatial processing" [73]
- Section 2.16, "Mobile-source processing with MOVES" [86]
- Section 2.19, "Creating model-ready emissions using MOVES lookup tables" [93]

Processing mobile sources involves a number of concepts that are unique to mobile sources. These include a special classification of road types in MOVES, SMOKE and MOVES vehicle types, emissions processes, MOVES emission factors, reference counties, reference fuel months, and meteorological processing using **Met4moves**. The following subsections explain these topics in more detail.

2.8.4.1. Special approach for on-road mobile MOVES SCCs

SMOKE handles SCCs differently for on-road mobile sources compared with all other source categories. SMOKE programs assume that on-road mobile SCCs have the following form:
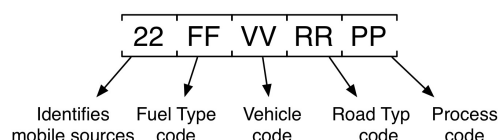
2.8.4.2. Fuel types in MOVES

MOVES can model six different fuel types: Gasoline, Diesel, Compressed Natural Gas (CNG), Liquefied Petroleum Gas (LPG), Ethanol, and Electricity. Table 2.2, "MOVES Fuel Type" [37] indicates a list of original MOVES fule types. Table 2.3, "Example of aggregated fuel types and corresponding MOVES fuel types" [38] shows how the four aggregated SCC fuel types are mapped to the six MOVES fuel types.

Table 2.2. MOVES Fuel Type

MOVES Fuel Type	Description
01	Gasoline
02	Diesel
03	Compressed Natural Gas (CNG)
04	Liquefied Petroleum Gas (LPG)
05	Ethanol (E-85)
09	Electricity

Table 2.3. Example of aggregated fuel types and corresponding MOVES fuel types

SCC	Example of aggregated Fuel Type Description	MOVES Fuel Type
x1	All non-diesel fuels	01; 03; 04; 05; 09
x2	All gasoline and ethanol blends	01; 05
x3	All fossil fuels	01; 02; 03; 04; 05
00	All fuels	01; 02; 03; 04; 05; 09

2.8.4.3. MOVES Vehicle Types

The vehicle types used in SMOKE's on-road mobile source processing are described in Table 2.4, "MOVES Vehicle type codes and descriptions" [38]. Table 2.5, "Example of aggregated vehicle types and corresponding MOVES vehicle types" [39] shows how the eight aggregated MOVES vehicle types are mapped to the original MOVES vehicle types.

Table 2.4. MOVES Vehicle type codes and descriptions

MOVES Vehicle Type	Description
11	Motorcycle
21	Passenger Car
31	Passenger Truck
32	Light Commercial Truck
41	Intercity Bus
42	Transit Bus
43	School Bus
51	Refuse Truck
52	Single Unit Short-haul Truck
53	Single Unit Long-haul Truck
54	Motor Home
61	Combination Short-haul Truck
62	Combination Long-haul Truck

Table 2.5. Example of aggregated vehicle types and corresponding MOVES vehicle types

SCC	Example of aggreg-ated Vehicle Type Description	MOVES Vehicle Type
30	Light Duty Trucks	31; 32
40	Buses	41; 42; 43
70	All Heavy Duty Trucks and Buses	41; 42; 43; 51; 52; 53; 54; 61; 62
71	All Heavy Duty Trucks	51; 52; 53; 54; 61; 62
72	All Combination Trucks	61; 62
80	All Trucks and Buses	31; 32; 41; 42; 43; 51; 52; 53; 54; 61; 62
81	All Trucks except Buses	31; 32; 51; 52; 53; 54; 61; 62
00	All Vehicles	11; 21; 31; 32; 41; 42; 43; 51; 52; 53; 54; 61; 62

MOVES can produce emission factors for possible combinations between 13 MOVES vehicle types and fuel types. For the efficiency of SMOKE processing, the emission factors can be aggregated to a list of vehicle types listed in Table 2.5, "Example of aggregated vehicle types and corresponding MOVES vehicle types" [39].

2.8.4.4. MOVES Road Types

MOVES can model nine different road types: rural restricted/unrestricted access with ramps, rural restricted/unrestricted access without ramps, urban restricted/unrestricted access with ramps, urban restricted/unrestricted access without ramps, and off-network (Table 2.6, "Road class and corresponding MOVES road type" [39]). Table 2.7, "Example of aggregated road types and corresponding MOVES road types" [40] indicates how the 6 aggregated MOVES road classes are mapped to the nine MOVES road types.

Table 2.6. Road class and corresponding MOVES road type

MOVES Road Type	Description
01	Off-Network
02	Rural Restricted Access
03	Rural Unrestricted Access
04	Urban Restricted Access
05	Urban Unrestricted Access
06	Rural Restricted without Ramps
07	Urban Restricted without Ramps
08	Rural Restricted only Ramps
09	Urban Restricted only Ramps

Table 2.7. Example of aggregated road types and corresponding MOVES road types

SCC	Example of aggregated Road Type Description	MOVES Road Type
70	Freeway	2; 4
71	freeway except ramps	6; 7
72	Ramps	8; 9
80	Non-Freeway	3; 5
90	All On-network	2; 3; 4; 5
00	All on and off-network	1; 2; 3; 4; 5

2.8.4.5. MOVES Process Types

MOVES can model 14 different process types: including on-roadway and off-network emissions processes, for the selected pollutants. Off-network emission processes (e.g., parked engine-off, engine starts, and idling, and fuel vapor venting in MOVES (Table 2.8, "Process types and corresponding MOVES road type" [40]). Table 2.9, "Example of aggregated process types and corresponding MOVES process types" [41] indicates how the 14 aggregated MOVES process types are mapped to the 14 MOVES process types.

Table 2.8. Process types and corresponding MOVES road type

MOVES Process	Description
01	Running Exhaust
02	Start Exhaust
09	Brakewear
10	Tirewear
11	Evaporative Permeation
12	Evaporative Fuel Vapor Venting
13	Evaporative Fuel Leaks
15	Crankcase Running Exhaust
16	Crankcase Start Exhaust
17	Crankcase Extended Idle Exhaust
18	Refueling Displacement Vapor Loss
19	Refueling Spillage Loss
90	Extended Idle Exhaust
91	Auxiliary Power Exhaust
99	Well-to-Pum>Well-to-Pumpp

Table 2.9. Example of aggregated process types and corresponding MOVES process types

SCC	Example of aggregated Process Type Description	MOVES Process Type
50	All Exhaust	1; 2; 15; 16; 17; 90; 91
51	All Exhaust except Ho-telling	1; 2; 15; 16
52	All hotelling exhaust	17; 90; 91
53	All Extended Idle Exhaust	17; 90
60	All Evaporative and Refuel-ing	11; 12; 13; 18; 19
61	All Evaporative except Re-fueling	11; 12; 13
61	All Evaporative except Re-fueling	11; 12; 13
62	All Refueling	18; 19
63	All Evaporative except Per-meation and Refueling	12; 13
70	All Exhaust and Evaporative and Refueling	1; 2; 11; 12; 13; 15; 16; 17; 18; 19; 90; 91
71	All Exhaust and Evaporative except Refueling	1; 2; 11; 12; 13; 15; 16; 17; 90; 91
72	All Exhaust and Evaporative except Refueling and Ho-telling	1; 2; 11; 12; 13; 15; 16
80	All Exhaust and Evaporative and Brake and Tire Wear except Refueling	1; 2; 9; 10; 11; 12; 13; 15; 16; 17; 90; 91
81	All Exhaust and Evaporative and Brake and Tire Wear except Refueling and Ho-telling	1; 2; 9; 10; 11; 12; 13; 15; 16; 17
00	All Processes	1; 2; 9; 10; 11; 12; 13; 15; 16; 17; 18; 19; 90; 91

2.8.4.6. Reference Counties

The approach for running MOVES for SMOKE relies on the concept of reference counties. These are counties that are used during the creation and use of emission rates to represent a set of similar counties (i.e., inventory counties) called a county group. The purpose of the reference county approach is to reduce the computational burden of running MOVES on every county in your modeling domain. By using a rep-resenative county, the user generates key emission rates for the single county in MOVES and then utilizes these factors to estimate emissions for all counties in the county group through SMOKE. The reference county is modeled at a range of speeds and temperatures to produce emission rate lookup tables (grams/mile or grams/vehicle/hour, depending on mobile emission process). The variables that are assumed to be constant across the county group members (and the reference county) are fuel parameters, fleet age distribution and inspection/maintenance (I/M) programs. The variables that can vary within the county group are vehicle miles traveled (VMT), source type vehicle population, roadway speed, and grid cell temperatures. Determ-

ining the reference counties and their respective county groups is a key aspect of utilizing the SMOKE-MOVES tool. It is ideal for the user to create each county group based on the similarity between the county characteristics (e.g., urban and rural) and the meteorological conditions (e.g., temperature and relative humidity). The user should avoid grouping counties that have significantly different meteorological conditions.

2.8.4.7. Reference Fuel month

Along with the concept of reference county approach, the concept of a fuel month is very important. It is used to indicate when a particular set of fuel properties should be used in a MOVES simulation. Similar to the reference county, the fuel month reduces the computational time of MOVES by using a single month to represent a set of months. To determine the fuel month and which months it corresponds to, the user should review the State-provided fuel supply data in the MOVES database for each reference county. If the fuel supply data change throughout the year, then group the months by fuel parameters. For example, if the grams/mile exhaust emission rates in January are identical to February's rates for a given reference county, then use a single fuel month to represent January and February. In other words, only one of the months needs to be modeled through MOVES.

2.8.4.8. Meteorological Data Processing

The meteorological data processor program **Met4moves** prepares spatially and temporally averaged temperatures and relative humidity data to set up the meteorological input conditions for MOVES and SMOKE using the Meteorology-Chemistry Interface Processor (MCIP) output files.

Met4moves must be run after MCIP and before the MOVES Driver script **"Runspec_generator.pl"** and SMOKE modeling system.

The following are the major processing steps that **Met4moves** performs:

- Read the reference county cross-reference file MCXREF that contains a list of reference counties and the county groups that map to those reference counties.

- Read the surrogate description file SRGDESC and a list of associated spatial surrogate(s) chosen for use in selecting grid cells.

- Determine a list of grid cells for each county. Only the selected grid cells are used to estimate the min/max temperatures, 24-hour temperature profiles, and RH over the user-specified modeling period.

- Set the dates of the modeling episode in local time using the flags STDATE and ENDATE

- Determine the fuel month for the reference county using the MFMREF input file.

- Read the country/state/county COSTCY or GEOCODE_LEVEL[1-4] (if USE_EXP_GEOCODES Y) file to define the time zones for county groups.

- Read the meteorology data that have been processed by MCIP.

- Calculate the min/max temperatures hourly and over the modeling period.

- Calculate average RH for the specified hour range over the modeling period.

- Once min/max temperatures and averaged RH are estimated for all reference counties and all inventory counties in the county groups, estimate diurnal 24-hour temperature profiles for use by the MOVES Driver script. The result is a normalized 24-hour shape profile over the user-specified period or fuel month.

2.8.4.9. MOVES Emission Processes by Emissoin Rate Tables

When the MOVES model runs for SMOKE, it runs for all emissions processes (or modes), including on-roadway and off-network emissions processes, for the selected pollutants. Off-network emission processes (e.g., parked engine-off, engine starts, and idling, and fuel vapor venting) in MOVES are hour-dependent due to vehicle activity assumptions built into the MOVES model; the emission rate depends on both hour of the day and temperature. On-roadway emission processes (e.g., running exhaust, crankcase running exhaust, brake wear, tire wear, and on-road evaporative), on the other hand, do not depend on hour. In MOVES, these emission processes are categorized into three major groups:

- **RatePerDistance (RPD) -** The emission rate of on-roadway vehicles (i.e., driving) from MOVES. The rate is expressed in grams/mile traveled.

- **RatePerVehicle (RPV) -** The emission rate of vehicles off-network (e.g., idling, starts, refueling, parked) from MOVES. The rate is given in grams/vehicle/hour.

- **RatePerProfile (RPP) -** The emission rate of vehicles off-network specifically, the evaporation from parked vehicles (vapor-venting emissions) from MOVES. The rate is expressed in grams/vehicle/hour.

- **RatePerHour (RPH) -** The emission rate of extended idle exhaust from on-roadway vehicle. The rate is expressed in grams/hour traveled.

MOVES emission rates are organized into four lookup tables (RPD, RPV, RPP, RPH), depending on emission process and whether the vehicle is parked or in motion. The approach to running MOVES for SMOKE is unique for each emission rate table listed in Table 2.10, "MOVES Emission Processes by Emission Rate Tables" [43]. A complete inventory must use the emission rates from all three tables. Note that refueling emission process is not a subject to MOVES emission rate table approach yet.

Table 2.10. MOVES Emission Processes by Emission Rate Tables

MOVES Lookup Table	Units	Emissions Process
RatePerDistance (RPD)	Grams/mile	Running Exhaust Crankcase Running Exhaust Tire Wear Brake Wear On-road Evaporative Permeation On-road Evaporative Fuel Leaks On-road Evaporative Fuel Vapor Venting
RatePerVehicle (RPV)	Grams/vehicle/hour	Start Exhaust Crankcase Start Exhaust Off-network Evaporative Permeation Off-network Evaporative Fuel Leaks Crankcase Extended Idle Exhaust Extended Idle Exhaust
RatePerProfile (RPP)	Grams/vehicle/hour	Off-network Evaporative Fuel Vapor Venting
RatePerHour (RPH)	Grams/hour	On-roadway Extended Idle Exhaust

The RPD lookup table is used to provide estimates of on-roadway emissions processes from mobile sources, using a separate file for each reference county. The on-road running processes that appear in this table include running exhaust, crankcase running exhaust, brake wear, tire wear, on-road evaporative permeation, on-road evaporative fuel leaks, and on-road evaporative vapor venting. The units of the emission rates in this table are grams/mile. The lookup fields for the factors are temperature and average speed. There are 16 set speed bins defined in Table 2.11, "MOVES Default Speed Bins" [45] (i.e., avgSpeedBinID 1=2.5mph, 2=5mph, 3=10mph, …16=75mph). The avgBinSpeed is used for interpolation in the RPD table.

The RPV lookup table is used to provide estimates of off-network emission processes (parked engine-off, engine starts, and idling), except for the evaporative off-network vapor venting emissions process. A separate file is provided for each reference county. The off-network emission processes include start exhaust, crankcase start exhaust, off-network evaporative permeation, off-network evaporative fuel leaks, extended idle exhaust, and crankcase extended idle exhaust. Fuel month, temperature, and local hour are the lookup fields in this table, and hours are in the local time of the modeling county. The units of the emission rates are grams/vehicle/hour. Note: Although the units are grams/vehicle/hour, the number of vehicles (i.e., population) should not be temporally allocated to hours in SMOKE. Instead, a county total of vehicle population should be multiplied by emission rates at any given hour. The number of starts per vehicle by hour is already accounted for in the MOVES lookup table.

The RPP table is used only to estimate emissions for off-network fuel vapor venting when the vehicle is parked. This process type includes diurnal (when the vehicle is parked during the day) and hot soak (immediately after a trip when the vehicle parks) emissions types. The process depends on the rate of rise in temperature and the maximum temperature achieved during the day for the diurnal emissions type, and on the hourly temperatures for the hot soak emission type. The lookup fields for this table are reference fuel month and hour of day. As with the RPV table, the units of the emission rates are grams/vehicle/hour. The estimated emissions rates need to be multiplied by the county vehicle population. The reference county lookup tables contain 24-hour emission rates per hour per vehicle using a reference county temperature profile with different minimum and maximum temperatures. The average day county emissions are determined by interpolating between the minimum and maximum temperatures for the modeling county generated by **Met4moves**. Section 2.8.4.8, "Meteorological Data Processing" [42] summarizes how **Met4moves** processes meteorological data for both MOVES and SMOKE.

The RPH table is used only to estimate emissions for on-roadway extended idle exhaust process from mobile sources, using a separate file for each reference county.

Table 2.11. MOVES Default Speed Bins

avgSpeedBinId	avgBinSpeed	AvgSpeedBinDesc
1	2.5	speed < 2.5mph
2	5	2.5mph ≤ speed < 7.5mph
3	10	7.5mph ≤ speed < 12.5mph
4	15	12.5mph ≤ speed < 17.5mph
5	20	17.5mph ≤ speed < 22.5mph
6	25	22.5mph ≤ speed < 27.5mph
7	30	27.5mph ≤ speed < 32.5mph
8	35	32.5mph ≤ speed < 37.5mph
9	40	37.5mph ≤ speed < 42.5mph
10	45	42.5mph ≤ speed < 47.5mph
11	50	47.5mph ≤ speed < 52.5mph
12	55	52.5mph ≤ speed < 57.5mph
13	60	57.5mph ≤ speed < 62.5mph
14	65	62.5mph ≤ speed < 67.5mph
15	70	67.5mph ≤ speed < 72.5mph
16	75	72.5mph ≤ speed

2.8.4.10. MOVES Pollutant Groups

The following Table 2.12, "MOVES Pollutant Groups" [46] provides a list of available MOVES pollutant groups that the user can specify to model within MOVES. The choice of pollutant groups(s) determines what pollutants are included in the three emission rate lookup tables (RPD, RPV, and RPP) output by MOVES. The letter 'X' marks the key pollutants for inclusion, and a letter 'd' signifies that the pollutant is included as a default in the MOVES run because a key pollutant depends on it. The user modifies the control.in input file to specify the pollutant group.

Table 2.12. MOVES Pollutant Groups

pollutantID	pollutantName	Pollutant Group			
		Ozone	Toxics	PM	GHG
1	Total Gaseous Hydrocarbons	d	d	d	
79	Non-Methane Hydrocarbons	d	d	d	
80	Non-Methane Organic Gases	d	d	d	
86	Total Organic Gases	X	X	X	
87	Volatile Organic Compounds	X	X	X	
2	Carbon Monoxide (CO)	X			
3	Oxides of Nitrogen	X		X	
30	Ammonia (NH3)			X	
32	Nitrogen Oxide	X		X	
33	Nitrogen Dioxide	X		X	
31	Sulfur Dioxide (SO2)			X	
100	Primary Exhaust PM10 - Total		d	X	
101	Primary PM10 - Organic Carbon		d	X	
102	Primary PM10 - Elemental Carbon		d	X	
105	Primary PM10 - Sulfate Particulate		d	X	
106	Primary PM10 - Brakewear Particulate			X	
107	Primary PM10 - Tirewear Particulate			X	
110	Primary Exhaust PM2.5 - Total			X	
111	Primary Exhaust PM2.5 - Organic Carbon			X	
112	Primary Exhaust PM2.5 - Elemental Carbon			X	
115	Primary Exhaust PM2.5 - Sulfate Particulate			X	
116	Primary Exhaust PM2.5 - Brakewear Particulate			X	
117	Primary Exhaust PM2.5 - Tirewear Particulate			X	
91	Total Energy Consumption		d	d	X
92	Petroleum Energy Consumption				X
93	Fossil Fuel Energy Consumption				X
5	Methane (CH4)	d	d	d	X
6	Nitrous Oxide (N2O)				X
90	Atmospheric CO2				X
98	CO2 Equivalent				X
20	Benzene		X	X	

pollutantID	pollutantName	Pollutant Group			
		Ozone	Toxics	PM	GHG
21	Ethanol				
22	MTBE		X		
23	Naphthalene		X		
24	1,3-Butadiene		X		
25	Formaldehyde		X		
26	Acetaldehyde		X		
27	Acrolein		X		

2.8.5. Point-source processing

Point-source emissions processing in SMOKE focuses on converting annual, daily, or hourly emissions to hourly, gridded model-ready emissions of the chemical species used by an AQM. Recall that by "point sources" in SMOKE we mean point sources in the usual sense plus wildfires with/without precomputed plumes. SMOKE processing may be performed either with or without growth and control of emissions. SMOKE can process both criteria and toxics inventories for point sources and combine consistent criteria and toxics inventories in one run (as explained in more detail in Section 2.9.5, "Combine toxics and criteria inventories" [55]).

Point-source processing can be performed using a CMAQ-based approach or a UAM-based approach, as previously described in Section 2.5.3, "Model-ready files" [21]. The processing steps for CMAQ base-year processing are shown in Figure 2.15, "Base case point-source processing steps for the CMAQ-based approach" [47]. In Figure 2.4, "Parallel approach to emissions processing" [27], we also included the major intermediate vectors and matrices; please refer to that diagram for those details.

Figure 2.15. Base case point-source processing steps for the CMAQ-based approach
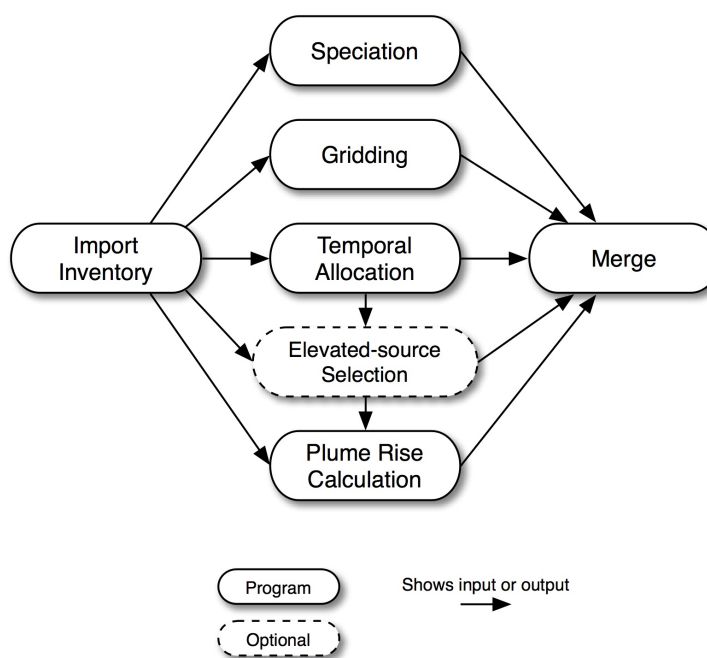
The inventory import step reads the raw emissions data, screens them, processes them, and converts the data to the SMOKE intermediate inventory file (inventory vectors in Figure 2.4, "Parallel approach to emissions processing" [27]). The import can optionally include day-specific and hour-specific data. The emissions in the inventory file are subdivided to hourly emissions during temporal allocation; assigned chemical speciation factors during speciation, and assigned spatial allocation factors during gridding. The plume-rise computation estimates vertical plume rise of emissions sources and computes the fraction of emissions from the sources to go into the model layers. The results of these steps are combined in a merge step, which creates model-ready files for CMAQ or MAQSIP.

Users may optionally choose to select specific sources to be elevated sources and/or PinG sources. If this approach is taken, the selection process can depend on daily-total emissions summed from the **Temporal** output files. Hence, Figure 2.15, "Base case point-source processing steps for the CMAQ-based approach" [47] shows that the elevated-source selection may optionally depend on the output from the **Temporal** program. If elevated-source selection is being included, the plume-rise computation uses that information to skip the point sources that have not been selected as elevated. Thus, plume rise is only computed for the elevated sources. The elevated-source selection also provides its results to the merge step, which is where the special PinG data files for CMAQ are created in addition to the 3-D model-ready file.

Figure 2.16, "Base case point-source processing steps for the UAM-based approach" [48] describes base-case processing for the UAM-based approach. For this type of modeling, the elevated-source selection step is required, and the plume rise computation is not performed. Otherwise, the major processing steps are the same as for the CMAQ-based approach.

Figure 2.16. Base case point-source processing steps for the UAM-based approach
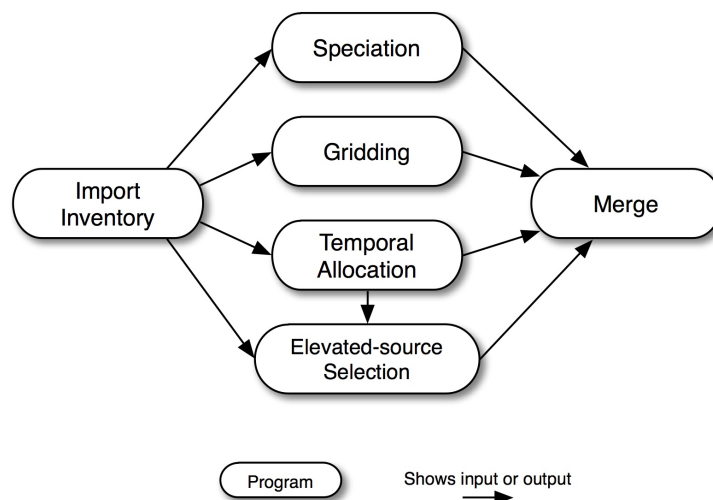
In Figure 2.17, "Future- or past-year growth and control point-source processing steps for the CMAQ-based approach" [49], we show the point-source CMAQ-based processing steps for future- or past-year processing. This processing is similar to the base-year processing flow, except the growth and controls step in added to calculate the growth and control matrices. The grow inventory step is added to convert the inventory from the base year to a future or past year. The control matrix can optionally be merged to apply control factors to the future- or past-year emissions. The steps shown with dotted lines represent steps that can be reused from the base-year processing because they do not necessarily depend on any of the new steps. However, if the elevated-source selection is to be performed based on the grown emissions, then the elevated-source selection and plume rise computation steps would need to be redone. Note that usually the same elevated-source list is used in both the base- and future-year modeling.

Figure 2.17. Future- or past-year growth and control point-source processing steps for the CMAQ-based approach
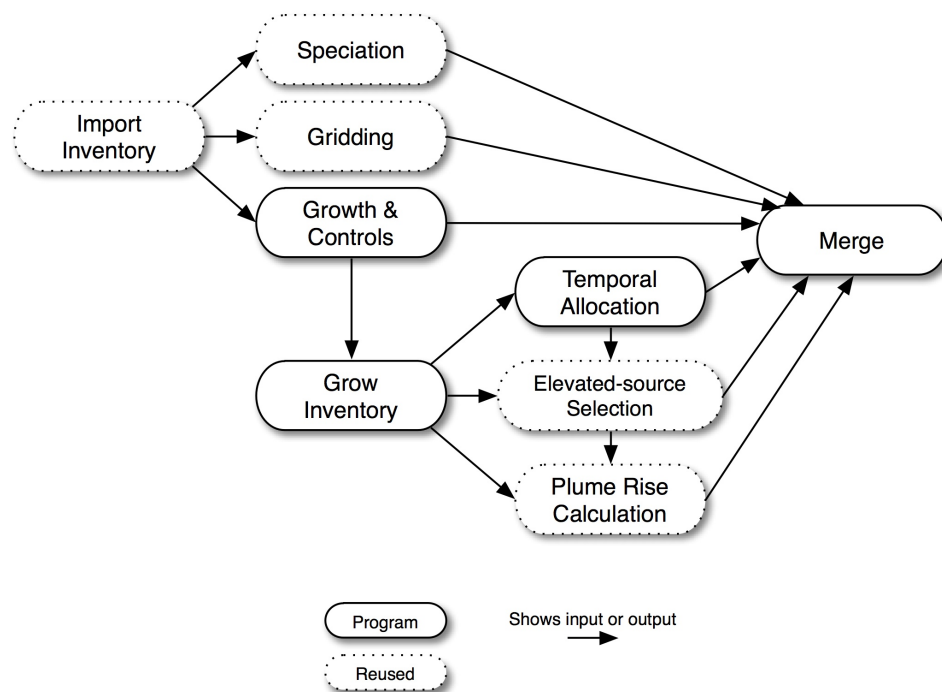
Growth and control can also be used for UAM-based processing (Figure 2.18, "Future- or past-year growth and control point-source processing steps for the UAM-based approach" [50]). As with the other figures, the dotted lines indicate steps that may be reused from the base-case processing. If the elevated-source selection depends on the grown emissions then you will need to regenerate the elevated-source list, though it is the usual practice in modeling to use the same elevated-source list in both the base- and future-year modeling. This allows the air quality modeling results to be more comparable.

Figure 2.18. Future- or past-year growth and control point-source processing steps for the UAM-based approach
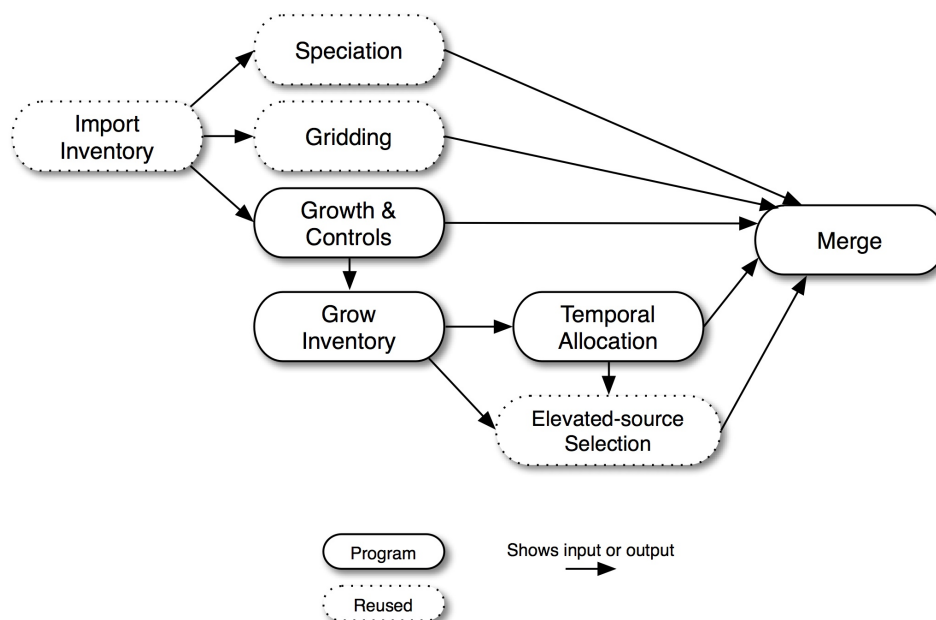
As with area sources, you may apply many growth and control matrices at the front end of processing. The area-source diagram for this approach was provided as Figure 2.10, "Alternative future- or past-year growth and control area-processing steps" [32], and the point-source approach is quite similar, with the addition of the elevated-source selection and plume rise computation steps.

In sections later in this chapter, we describe the SMOKE programs that are needed for each of the processing steps just discussed and additional details about what activities are accomplished during each step. These sections are:

*   Section 2.9, "Inventory import" [53]
*   Section 2.10, "Temporal processing" [63]
*   Section 2.11, "Chemical speciation processing" [70]
*   Section 2.12, "Spatial processing" [73]
*   Section 2.13, "Growth processing" [77]
*   Section 2.14, "Control processing" [80]
*   Section 2.15, "Elevated-source processing" [83]
*   Section 2.18, "Creating model-ready emissions" [92]
*   Section 2.20, "Quality assurance" [94]

Point-source processing includes a number of additional features that are not applicable for other SMOKE source categories:

1. Flexible source definitions
2. Stack parameters
3. Day- and hour-specific emissions
4. Different approaches for elevated sources for different AQMs, including the use of PinG sources
5. Elevated-source selection
6. Wild and prescribed fires point sources

2.8.5.1. Flexible source definitions

Depending on the input format of the point-source emissions (e.g., FF10, ORL), the set of characteristics that are used to uniquely identify a point source can be different. For example, the ORL-formatted inventories define a point source using a country, state, and county code, an SCC, a plant identifier, a stack number, a point identifier, and a segment number. The FF10 format, however, identifies a source using a FIPS state/county code, a plant code, a unit ID, a segment ID, and SCC. To better support the formats and be adaptable if new formats are created in the future, SMOKE uses a flexible definition of point sources. This definition consists of the following source characteristics to uniquely define the sources:

- Country, state, and county code

- Plant ID (15 characters or less)

- Characteristics 1 through 5 (each 15 characters or less)

Depending on the input format, SMOKE assigns different variables from the input format to the parts of the SMOKE point-source definition. The assignments for the remaining characteristics are as follows:

ORL format:

- Char 1: Point ID
- Char 2: Stack ID
- Char 3: Segment ID
- Char 4: SCC
- Char 5: unused

FF10 format:

- Char 1: Stack ID
- Char 2: Unit ID
- Char 3: Segment ID
- Char 4: SCC
- Char 5: unused

The meaning of these source characteristics for a given inventory type needs to be considered when cross-reference files are created, if cross-reference entries other than state/county and SCC-specific entries are provided.

2.8.5.1.1. Point definition header row in cross-reference files

As just described, SMOKE uses a flexible definition of point sources. This definition may or may not include the SCC (although SCC is always at least a source attribute). Cross-reference files for point sources can contain source-specific records, and they usually use the SCC to perform the needed assignments during emissions processing. For the files to be self-describing, they use a header that indicates the number of characteristics in addition to the plant ID that are being used in the cross-reference file. This number needs to be consistent with the number of point-source characteristics used in the inventory files. In addition, the header indicates which, if any, of the point-source characteristics is the SCC. This header starts with the characters /POINT DEFN/, and the files that use it describe it as part of the file format definition in Chapter 8, *SMOKE Input Files* [345].

2.8.5.2. Stack parameters

Several of the source attributes for point sources are stack parameters - specifically, the stack height, stack diameter, and the stack flue gas exit temperature, velocity, and flow rate. SMOKE can use hourly data for

the stack flue gas exit temperature, velocity, and flow rate when using the CMAQ-based approach to modeling with SMOKE computing hourly plume rise. The hourly stack parameters cannot be used when modeling using a UAM-based elevated-point-source approach.

During the **Smkinven** program's import of the stack parameters from the annual or average-day inventory file (i.e., not the hourly stack parameters), SMOKE needs to read or assign stack parameters for all point sources. Section 2.9.9, "Fill in and check point-source stack parameters" [58] explains in greater detail what the **Smkinven** program does with stack parameters. The hourly stack parameters are read in without modification or adjustment.

2.8.5.3. Day-specific and hour-specific emissions

Emissions from point sources are sometimes available as day- or hour-specific values. **Smkinven** can import the day- and hour-specific data, and it can also convert the hour-specific data to hour-specific temporal profiles. When these data are available, the **Temporal** program overrides the annual or daily emissions with the most specific data available. If day-specific data are available, **Temporal** uses them to overwrite the annual or average-day emissions during the time periods that these data are available. If hour-specific data are available, **Temporal** uses them to overwrite the annual, average-day emissions, or day-specific emissions data.

2.8.5.4. Different approaches for elevated sources for different AQMs

As introduced in Section 2.5.3, "Model-ready files" [21], there are two different major approaches for creating emissions inputs to AQMs: the CMAQ-based approach and the UAM-based approach. The two approaches differ only on how point sources are being treated. In the CMAQ-based approach, SMOKE calculates the plume rise using an algorithm based on a Briggs plume rise formulation. SMOKE then includes the vertical distribution of the point-source emissions in the 3-D model-ready file for CMAQ or MAQSIP. For the CMAQ model only, SMOKE can also create two special PinG files: one to identify the sources, their locations, and their stack parameters, and the other to provide the hour-specific emissions for just these sources. In the UAM-based approach, SMOKE creates a special elevated-point-source file that both identifies the elevated and PinG sources and includes the hourly emissions values for those sources.

PinG sources are those sources that will be treated in greater detail by the AQM. In simple terms, the AQMs preprocess the chemistry of the plume emissions before those emissions are provided to the AQM grid cells and layers. The intent of the PinG approach is to provide more accurate modeling at and around very large point sources.

Section 2.15, "Elevated-source processing" [83] describes in greater detail the steps taken by SMOKE in the layer fraction processing using a Briggs formulation and the elevated and PinG source selection.

2.8.5.5. Wild and prescribed fires point sources

You may also either provide either precomputed point-source plume rise to SMOKE or internally compute plume rise using acres burned and fuel loading of fires with both the CMAQ-based and UAM-based approaches to modeling point sources. Precomputed plume rise point sources are called explicit plume sources. This capability was implemented for modeling wildfire sources as point sources in SMOKE using plume rise computed with a different approach from the Briggs-based approach used for stack-based plumes (Section 4.4.17, "Plume Rise Calculation for Fires" [167]). The input data for this approach are the fraction of emissions in layer 1, the bottom of the plume, and the top of the plume. SMOKE distributes the emissions across the layers by weighting the emissions by the pressure difference in each layer over the total pressure difference between the top and bottom of the plume.

For the UAM-based modeling approach, the file format does not readily allow you to provide precomputed plume rise; in fact, the entire premise of the format is that the AQM will compute the plume rise. To enable

you to provide precomputed rise for the UAM-based modeling approach, the **Smkmerge** program creates an ASCII elevated-point-source file with an imaginary stack for each layer of each source (e.g., each wildfire). The stack parameters of the imaginary stack are set to values that will ensure a zero plume rise will be computed for the stack, and the x-y location of the stacks are the same for all imaginary stacks representing the same source. The emissions associated with the imaginary stacks are provided based on the emissions values that are to be entered in each layer for the source. The emissions for layer 1 are written in the point-source file to the imaginary stack associated with layer 1, and the same is done for all of the other layers. SMOKE uses a zero value for the imaginary stack when the emissions from a given source are not in a layer for an hour or when the source stops (e.g., once a wildfire ends). While not particularly elegant, this approach permits providing precomputed plume rise to the UAM-based models without having to change those models.

2.9. Inventory import

The importing of emission inventory and related data is the first processing step needed for any emissions processing effort. The **Smkinven** program imports data for anthropogenic sources, and the **Normbeis3** program imports BEIS3 land use data for biogenic sources. In this section, we focus on the import of the anthropogenic inventories using **Smkinven**. The biogenic import is further described in Section 2.17, "Biogenic processing" [91].

Smkinven performs many types of activities during import of the anthropogenic inventories. Though the primary purpose is reading the data from ASCII formats and outputting and I/O API SMOKE intermediate inventory, there are many other actions that need to be performed duringthe inventory import stage of processing. These actions are the following:

1. Check that the formats of the input files are correct and consistent, and ensure that all data can be read properly.

2. Assign pollutant names to data input by code numbers.

3. Select pollutants from the input files to keep for further SMOKE processing.

4. When multiple files are provided, combine all annual and/or average-day data into aconsistent inventory. This includes checking for duplicates and possibly aborting, depending on program options set by the user.

5. Combine toxics and criteria inventories, and eliminate duplicate mass using either an integrate or no-integrate approach.

6. Sort the inventory records into the order expected by other SMOKE programs.

7. Aggregate or disaggregate toxics emissions data as specified by user inputs.

8. Assign point-source locations to area sources, when available.

9. Fill in and check point-source stack parameters.

10. Convert stack locations from UTM to lat-lon.

11. Optionally ensure that lat-lon coordinates are in the Western Hemisphere.

12. Convert units of emissions and activities to the units used in the SMOKE intermediate inventory.

13. Set the weekday averaging approach.

14. Assign country codes and/or geographic codes (GEOCODE_LEVEL[1-4]), years, and time zones.

15. Handle inventories that have data for multiple years.

16. Set the base year.

17. Report results of import including pollutant totals for toxics data and other information needed for quality assurance.

18. Import day-specific and hour-specific data, if available, and ensure that records in these files match inventory records provided in the annual or average-day inputs.

In the following subsections, we describe what SMOKE does for each of these activities.

2.9.1. Check the correctness and consistency of input file formats

Smkinven can read the following ASCII formats for annual and average-day inventory data:

- **ORL format:** This set of input formats is used for inputting point, nonpoint, on-road, and nonroad HAP emissions inventories, also called toxics emission inventories. There is a different ORL format for nonpoint, point, nonroad mobile, and on-road mobile sources.

- **List format:** This is the input format used to provide multiple files to **Smkinven** in a single run. This format is simply an ASCII file that contains a list of other files.

- **List GRID format:** This is the input format used to provide multiple global gridded emission inventory files to **Smkinven** in a single run. This format is simply an ASCII file that contains a list of other files. To support input of multiple pre-gridded NetCDF files, the keyword #LIST GRID in the header of this file will switch SMOKE into gridded inventory processing mode. This approach is described more in Section 2.8.2.2, "" [32]

- **Gridded I/O API format:** This format is a gridded I/O API file for allowing the import of pregridded data from the same modeling domain. This approach is described more in Section 2.8.2.2, "" [32].

Smkinven ensures that all file formats provided to SMOKE are correct and include the required data fields. The formats and their required fields are provided in Chapter 8, *SMOKE Input Files* [345].

2.9.2. Assign pollutant names to data input by code numbers

The ORL format use code numbers (usually Chemical Abstracts Service [CAS] numbers) to distinguish which chemical compound or inventory pollutant is provided on each line of the file. **Smkinven** matches these numbers with the CAS numbers from an inventory table (`INVTABLE`) file, described in Section 8.10.4, "`INVTABLE`: Inventory table" [450]. The CAS number does not necessarily have to be a valid CAS number; it can be any number as long as there is a match between the numbers in the ORL file and the inventory table. The inventory table provides the inventory data names, such as the pollutant names that SMOKE uses in the remaining processing steps. Note that the SMOKE inventory pollutants may not be identical to the pollutants in the inventory fed to SMOKE because of the aggregation/disaggregation that is performed by **Smkinven** (see Section 2.9.7, "Aggregate or disaggregate toxics emissions" [57] for more information).

For toxics processing, if multiple inventory data names apply for the same CAS number, the Factor column of `INVTABLE` will contain the split factor used by **Smkinven** to disaggregate the emissions from that CAS

number to multiple inventory data values. If multiple CAS numbers apply for the same pollutant name, then **Smkinven** will sum these emissions, but will not report duplicate records unless there are indeed duplicates in the inventory file. This is described more fully in Section 2.9.7, "Aggregate or disaggregate toxics emissions" [57]. Duplicate reporting is described more in Section 2.9.4, "Check for duplicate records" [55].

The use of the inventory data name as the unique pollutant identifier in SMOKE differs from the approach of EMS-HAP, in which the SAROAD code is the unique identifier for the pollutants to be modeled. Because we anticipated that some toxics pollutants that do not have unique SAROAD codes (e.g., divalent particulate mercury) would need to be modeled explicitly, we did not want to take this approach. If we had, the user would have been required to create fake and unique SAROAD codes to be able to model these emissions explicitly.

2.9.3. Select pollutants from the input files for further SMOKE processing

In version 1.5 and higher of SMOKE, users can specify the valid data (pollutants and activities) using the inventory table discussed in Section 8.10.4, "INVTABLE: Inventory table" [450], as described in Section 2.9.2, "Assign pollutant names to data input by code numbers" [54]. **Smkinven** will read only those entries that have a "Y" (for "Yes") in the Keep column of the inventory table. If a pollutant has an "N" (for "No") in the Keep column, it will not be output to the SMOKE intermediate files, but will be included in the inventory reports that **Smkinven** creates. If the pollutant is not listed at all in the inventory table, it will be dropped as well; **Smkinven** will also write a warning message and will not include the pollutant in any reporting. It is a good work practice for users to put all pollutants in the input inventory in the inventory table even if it the pollutant will not be used for the AQM. Use of the INVTABLE file replaces the use of the SIPOLS and ACTVNAMS files from previous versions of SMOKE.

2.9.4. Check for duplicate records

In Section 2.9.1, "Check the correctness and consistency of input file formats" [54], we explained that **Smkinven** ensures that the input files are correct and that the same format is used. In addition, **Smkinven** checks for duplicate records across the entire set of inventory input files. A duplicate record is one that has the same source characteristics (defined in Section 2.8.1, "Summary of SMOKE processing categories" [29]) and pollutants as another record in the inventory. **Smkinven** provides an option that allows you to instruct **Smkinven** whether duplicates should cause a warning message or an error. In some cases, you may not expect duplicates in your inventory, in which case you can have **Smkinven** abort after reporting all duplicates. In other cases, you can instruct **Smkinven** to sum the emissions across all duplicate records. See Section 6.13, "**Smkinven**" [260] for more information on the different approaches.

2.9.5. Combine toxics and criteria inventories

For point, nonpoint, on-road mobile, and nonroad-mobile sources, the toxics inventory contains emissions for VOC pollutants that are provided as explicit chemical compounds (for example, benzene). These same VOC emissions are also included as an aggregated VOC value in the criteria emissions inventory. To use these inventories together, **Smkinven** provides the necessary options to ensure that double counting of VOC emissions will not occur. These two options are the "integrate" and "no-integrate" options.

The "integrate" option involves subtracting toxic VOC emissions from the criteria VOC emissions to avoid double counting of VOC when the emissions are speciated. With this option the user must ensure that the sources in the toxics and criteria inventories match up one-to-one, so that **Smkinven** can properly compute the emissions.

Note that in any discussion of the toxics inventory we have assumed that all emissions are annual total emissions, because the toxics inventory that is currently available does not include average-day emissions. We have also assumed that the inventory contains VOC emissions, but the same approach can be used to process TOG emissions.

During import of both toxics and criteria emission inventories, SMOKE matches the area/nonpoint, on-road mobile, and nonroad mobile emission inventories by country/state/county code and SCC. SMOKE also matches the toxics and criteria records for the point sources, provided that the point sources in the two inventories use identical fields for their source characteristics. You are required to ensure that the source characteristics for all source categories match between the two inventories for any sources that you wish to have matched. Once they are matched, SMOKE will have both a criteria VOC emissions value and toxics emission values for individual VOC chemical compounds.

SMOKE can optionally compute a NONHAPVOC value by subtracting the sum of toxics VOC from the criteria VOC value. This same approach can be used to create a NONHAPTOG value if the inventory or MOVES (when processing on-road mobile emissions using VMT data) uses a TOG value instead of a VOC value. (We will not mention NONHAPTOG again, but it could be used to replace NONHAPVOC throughout this section).

The case of computing NONHAPVOC is called the "integrate" case because it involves integration of the VOC mass between the criteria and toxics inventories. Likewise, the case of not computing NONHAPVOC is called the "no-integrate" case. With the "integrate" approach, the NONHAPVOC mass and the toxics VOC mass are independent from one another and will not double count emissions. The calculation must be performed for each source, and **Smkinven** will set the criteria VOC value to zero when it computes the NONHAPVOC value. **Smkinven** determines which pollutants should be subtracted from VOC using the "VOC or TOG" column in the inventory table (INVTABLE) file.

2.9.6. Sort the inventory

When **Smkinven** reads an inventory, it also puts its sources into a special sorted order prior to outputting the SMOKE intermediate inventory files. All programs that read **Smkinven** outputs, which includes most of the SMOKE programs, expect this order. The order is determined by sorting the source characteristics listed in Section 2.8.1, "Summary of SMOKE processing categories" [29] in ascending order. For example, area sources will be sorted in order of increasing country/state/county code, and within a single country/state/county code will be sorted in order of increasing SCC. In Figure 2.19, "Combining and sorting ASCII inputs to created sorted I/O API outputs" [57], we show how the sorted order may be completely different from the order of the files and records provided to **Smkinven**. The figure shows the unsorted ASCII input files at left (provided to **Smkinven** by logical files ARINV, MBINV, or PTINV) and how the records can be rearranged by **Smkinven** to create the sorted I/O API output files (output from **Smkinven** as AREA, MOBL, or PNTS). Each record in this diagram represents a complete inventory record with all source characteristics, source attributes, and emissions.

Figure 2.19. Combining and sorting ASCII inputs to created sorted I/O API outputs
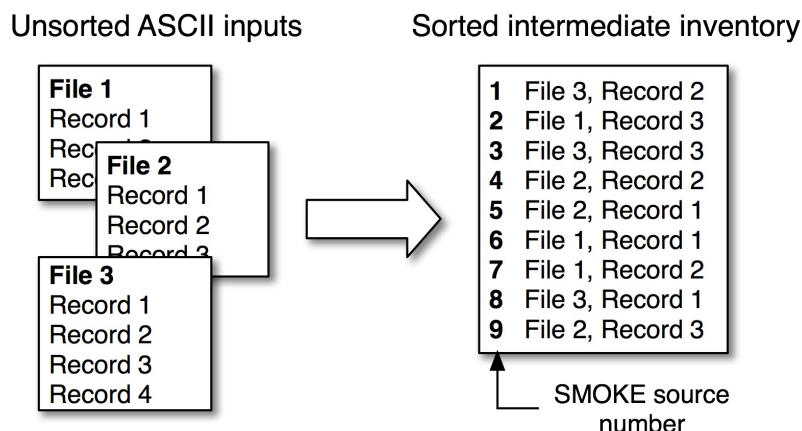
Having this sorted order is important because the programs that depend on **Smkinven** outputs also store their outputs in the same order. However, these other programs (such as **Temporal** for temporal allocation, **Spcmat** for chemical speciation, and **Grdmat** for spatial allocation) store their records using the record numbers to match their outputs with the **Smkinven** outputs. This means that the source characteristics that are stored in the **Smkinven** outputs are not also included in the outputs from **Temporal**, **Spcmat**, **Grdmat** and other programs; these programs rely on the sorted order in the **Smkinven** outputs not changing. This approach allows minimal redundant data storage, this reducing disk space needs.

As explained previously, the records output from **Smkinven** are vectors of emissions and source characteristics that make up the SMOKE intermediate inventory files. Each record number in the file identifies an element of the vector. The outputs from **Temporal** are also vectors of hourly emissions. The record number in each hourly vector will match the record number in the intermediate inventory files. The outputs from **Spcmat** are a matrix of speciation factors, in which the record numbers (rows of the matrix) will match the record numbers of the intermediate inventory files. The columns of the matrix are each valid pollutant-to-species transformations. The outputs from **Grdmat** are a sparse matrix, but again the rows of the matrix match the rows of the intermediate inventory file. Therefore, assignment of factors is a simple matter of selecting the same record number from the **Smkinven** output files; this is in fact one part of the vector-matrix multiplication used by SMOKE.

It is important to remember this sorted-order approach when you have run an inventory through all of the programs once, and then want to change your inventory and re-import the data with **Smkinven**. For the re-importing and subsequent rerun, if any source characteristics in the inventory change, or if any sources are added or removed, then the number and/or order of the output sources in the new **Smkinven** outputs will be different. This means that the outputs from all processing steps that depend on the **Smkinven** outputs will need to be rerun.

2.9.7. Aggregate or disaggregate toxics emissions

Smkinven also supports aggregation and disaggregation of toxics inventory pollutants to match the input needs for AQMs. There are two components of this aggregation and disaggregation. The first involves what to do during inventory import to resolve discrepancies between the inventory content and what data are most useful for further processing through SMOKE; this topic is addressed in this subsection. The second issue is conversion of the toxics inventory to the species needed by the AQM, which is handled by the **Spcmat** program in SMOKE, as described in Section 2.11, "Chemical speciation processing" [70].

Smkinven uses the inventory table (`INVTABLE`) file to determine what aggregation and disaggregation are needed. The `INVTABLE` file can have both multiple pollutant names per CAS number and multiple CAS numbers per pollutant name. In the first case, **Smkinven** disaggregates the emissions to two or more pollutants using the number provided in the "Factor" column of the `INVTABLE` file. In the second case, **Smkinven** combines the separate records in the inventory into the same pollutant. Depending on the relationship between the "Keep" column, the "Pollutant Name," and the "CAS Number" in the inventory table file, the reader routines ensure that they will keep emissions only for pollutants (not CAS numbers) with a "Y" in the "Keep" column. **Smkinven** can also tell the difference between two CAS numbers that are being aggregated for the same source and a duplicate record (two records for the same source with emissions for the same pollutant).

2.9.8. Assign point-source locations to area sources

Some area/nonpoint and nonroad mobile sources can be assigned point-source locations instead of being assigned spatial surrogates. At this time, the only allowable cases of this are for toxics processing for nonroad mobile and stationary area/nonpoint sources related to airport emissions processes. **Smkinven** has the capability to make such assignments for any area source from both the toxics and criteria inventories; the example file that comes with SMOKE for this purpose (the `ARTOPNT` file) only contains entries for the airport emissions processes. Section 4.2, "Test case descriptions" [117] provides more information about the example `ARTOPNT` file provided with SMOKE.

Because airport location data are readily available, this feature gives users the ability to model airport-related area-source emissions at airport locations, as opposed to spatially allocating them to grid cells using spatial surrogates. Although spatial surrogates could be (and have been) developed to reproduce the same or similar results as the point-source assignments for models such as CMAQ, other models that SMOKE may eventually support (namely ISCST3) require that point-source inputs include the location coordinates. The area-to-point assignment feature in SMOKE will support such a need. Note that for large airports and a small grid, one may want to have the airport emissions gridded using surrogate data since the airport can be large enough to encompass multiple grid cells. Thus, the user should decide whether and when to use this area-to-point feature as opposed to gridding using an airport surrogate.

In SMOKE version 1.5 and higher, SMOKE can assign an area source to one or more locations in a county. For example, when multiple airports are in a county, SMOKE adds sources to the inventory and splits the county-total emissions among the locations based on a factor provided in the `ARTOPNT` file. All sources remain with the area-source inventory in which they started; they are not moved to a point-source inventory. Any area sources that are assigned point locations are spatially allocated using those locations, as described in Section 2.12, "Spatial processing" [73]. Sources that are not assigned point-source locations are allocated in the standard way using spatial surrogates.

2.9.9. Fill in and check point-source stack parameters

An additional action taken by the **Smkinven** program for point sources only is to check stack parameters and fill them in with valid values if they are missing. **Smkinven** performs the following steps on stack parameters from the annual or average-day inventory files, but not the hour-specific stack parameters file.

- If the stack parameters are zero, **Smkinven** treats them as missing values and writes the warning messages.

- If the exit velocity is missing or zero and the exit flow is not, or if the `VELOC_RECALC` option is Y in the run script, **Smkinven** automatically calculates the exit velocity using the formula:

velocity = flow / (π * diameter^2/4)

- If the stack parameters are not missing or zero, **Smkinven** ensures that their values are within the allowed ranges. The range for each stack parameter is as follows:
  - Height: 0.5 to 2100 meters
  - Diameter: 0.01 to 100 meters
  - Exit temperature: 260 to 2000 K
  - Exit velocity: 0.0001 to 500 m/s

  When a stack parameter falls outside of its associated range, **Smkinven** sets it to the top or bottom of the range, depending on whether it is higher than the upper end of the range or lower than the lower end. Note that a zero value is not treated as an out-of-range parameter, but is treated as a missing value.

- If the stack parameters are missing or zero, **Smkinven** uses the PSTK file to assign new stack parameters using country/state/county and SCC assignments. The format of PSTK file is described in Section 8.9.3.2, "PSTK: Point source stack replacement parameters file" [443].

2.9.10. Convert coordinates from UTM to lat-lon

The ORL and FF10 point-source formats permit stack coordinates to be provided in UTM or lat-lon co-ordinates, and the SMOKE intermediate inventory stores the coordinates as lat-lon values. **Smkinven** converts the UTM coordinates to lat-lon coordinates using the I/O API routine UTM2LL [http://www.baronams.com/products/ioapi/LL2UTM.html]. Lat-lon coordinates must be provided in decimal degrees, while UTM coordinates must be provided in meters.

2.9.11. Optionally ensure that lat-lon coordinates are in the Western Hemisphere

For any lat-lon coordinates input to **Smkinven** (point-source locations, link coordinates, and area-to-point coordinates), SMOKE can optionally ensure that the longitude values are in the Western Hemisphere. In some cases, the data prepared for input to SMOKE do not contain the negative sign on the longitude value that indicates the Western Hemisphere. If the modeling domain is in the Western Hemisphere, then **Smkinven** will convert all positive longitudes to negative ones when the WEST_HSPHERE option is set to Y.

2.9.12. Convert units of emissions and activities

Different input formats in SMOKE have different emissions units. The SMOKE intermediate inventory file stores all annual and average-day emissions values from the ORL format in tons/year. These two formats also support average-day emissions values, which are stored separately in the SMOKE intermediate inventory in tons/day. The average-weekday emissions values input are also stored in tons/year, but SMOKE sets an internal variable called TPFLAG so that later SMOKE processing steps can properly treat the computed "annual" value as an average-day value. The annual VMT data are stored as miles/year. In all cases, **Smkinven** converts the units of the input emissions to the units used in the SMOKE intermediate inventory. When this conversion involves a day-to-year conversion, **Smkinven** considers leap years by using 366 instead of 365 days in the year. See Section 8.2, "Inventory Files" [351] for information about the units required for each inventory format.

When emissions are provided as average-day values from the ORL and FF10 formats and these emissions are then used in later processing steps, SMOKE does not further adjust the emissions using the monthly profiles. SMOKE assumes that the average-day emissions from these two formats have been adjusted to a specific month already. In addition, when **Smkinven** is configured using the FILL_ANNUAL option to fill in missing annual values using average-day values, **Smkinven** sets the TPFLAG internal variable to indicate that monthly adjustments should not be applied and that the "annual" emissions should just be

divided by the number of days of the year before being used. **Smkinven** makes this setting on a source-by-source basis.

2.9.13. Set the weekday averaging approach

There are two approaches for processing weekly temporal profiles: weekly normalization and weekday normalization. Although the actual normalization happens during temporal allocation, **Smkinven** sets the approach to use for each source. In the weekly approach, the **Temporal** program normalizes the weekly temporal profiles over every day of the week. This approach is appropriate for annual-total inventories or average-day inventories. Weekday normalization normalizes over just the weekdays (i.e., Monday through Friday) in the profile, or if the profile indicates no emissions on weekdays, then over just the weekend days in the profile. This approach is appropriate only for average-weekday inventories. Section 2.10, "Temporal processing" [63] describes the two forms of weekly profile normalization in more detail.

The default normalization setting for ORL and FF10 formats in **Smkinven** is weekly normalization. The default settings can be changed using the `WKDAY_NORMALIZE` option in the SMOKE run scripts. When this option is set to Y, the sources will be set to instruct **Temporal** to use weekday normalization, and when this option is set to N, the sources will be set to cause weekly normalization. **Smkinven** gives a warning when an approach is being used that is inconsistent with the expected approach for the inventory format being used.

2.9.14. Assign country codes, years, and time zones

For all source categories, **Smkinven** assigns country codes, years, and time zones to inventory files that do not contain this information in the inventory records.

Since most inventory formats do not include a column for country or year, the country code and year must be provided in header fields in the inventory. because most inventory formats do not include a column for country or year. Users can provide data for up to 10 countries using as many separate files, or separate headers within a single file, as needed. The inventory file formats listed in Section 8.2, "Inventory Files" [351] provide a further description of these country-setting headers and the valid country codes.

Users can also provide multiple inventory years in a single **Smkinven** run. This is necessary in some cases when an inventory for one region (e.g., Canada or Mexico) is unavailable for the same year as the majority of the inventory region (e.g., the U.S.). **Smkinven** stores the inventory year as part of the source attributes.

In addition to country codes and years, **Smkinven** assigns a time zone to each source based on the county associated with the source. **Smkinven** uses the `COSTCY` or the `GEOCODE_LEVEL4` (if USE_EXP_GEOCODES Y) file to get this information. It matches the county code from the inventory to the county code in this file. If a county code included in the inventory is missing from this file, or if the time zone is not provided in the `COSTCY` or `GEOCODE_LEVEL4` file, **Smkinven** uses the `SMK_DE-FAULT_TZONE` setting to obtain a default time zone for sources in such a county. In some cases, the time zone set in the `COSTCY` or `GEOCODE_LEVEL4` file is an approximation, since counties that are bisected by two time zones have only the predominant zone represented in this file.

2.9.15. Handle inventories that have data for multiple years

In some cases, inventories may use data from multiple years; this can occur when inventory data are not available for the modeling year of interest, but data from another year are available. SMOKE can handle this case. Unless growth factors are available to grow the available data to the desired year, the SMOKE intermediate inventory may include data from two years. Usually, using data from a nearby year is preferable to not performing the modeling at all.

2.9.16. Set the base year

Smkinven imports all data and then sets the inventory base year in the file as the year that has the most data values. The year of the inventories is set using the #YEAR header for ORL and FF10 inputs, (see Section 8.2, "Inventory Files" [351] for more information on this header and packet). Usually, the year of the data is the same for all sources, so you will know what **Smkinven** will assume is the base year. However, if multiple inventory years are provided, **Smkinven** will set it using the year that is associated with the largest number of sources.

If you provide a future-year inventory to SMOKE, you must override the base-year setting obtained from the values given by the #YEAR header or the /INVYEAR/ packet. This is accomplished with the SMK_BASEYR_OVERRIDE setting. This is necessary so that later SMOKE steps will be able to verify that the episode year is consistent with the base year.

2.9.17. Report results of import step

The **Smkinven** program writes several reports to help you determine what it has done.

1. A summary by CAS number of the emissions in tons/year. This includes the number of inventory records; whether all, some, or none of the pollutants associated with that CAS code are kept; and the CAS description. This report is written to a special report file (REPINVEN).

2. A summary of emissions by CAS number and pollutant before and after the application of disaggregation factors based on the inventory table. This report shows, for instance, how chromium emissions get split out into different pollutants. The report is written to the REPINVEN file.

3. A listing of the first 10 nonheader records in each inventory file. This information is written to the program log file.

4. For area-to-point conversions (from Section 2.9.8, "Assign point-source locations to area sources" [58]), a list of the SCCs that were converted, and the section number of the ARTOPNT input file used to assign the point locations. This information is written to the program log file.

5. For area-to-point conversions, a list of any SCCs included in the ARTOPNT input file that do not also appear in the inventory. This report is written to the REPINVEN file.

6. For area-to-point conversions, a summary of emissions totals by SCC and pollutant before and after the factors are applied, and the total number of country/state/county codes affected. A separate summary is provided that reports emissions by state, SCC, and pollutant, but otherwise has the same information. These reports are written to the REPINVEN file.

7. For area-to-point conversions, a summary by SCC of the number of country/state/county codes being assigned area-to-point factors, and the number not being assigned those factors. This report is written to the REPINVEN file.

8. For import of CEM data, a summary of ORIS IDs and boiler IDs from the CEM data that have been matched to the inventory. SMOKE allows for multiple plant descriptions and country/state/county codes for a single ORIS ID that may be identical in their characteristics but different boiler IDS. They are treated as separate sources by matching boiler IDs. This report is written to the REPINVEN file.

More information about these reports can be found in Section 10.5.1, "REPINVEN" [487].

2.9.18. Import day-specific and hour-specific data

Another function of **Smkinven** that only applies to point sources is importing day- and hour-specific data. It can read three input ASCII formats:

- **CEM data format:** The CEM data format provides standard local time hourly emissions by ORIS identification code and boiler number. It was developed in conjunction with the Market Trading Division of EPA, and that division now provides this ASCII file format when the data are released.

- **FF10 format:** The FF10 day- and hour-specific format can be input to SMOKE using either the FF10 annual or average-day inventories.

In addition to reading the files, **Smkinven** ensures that the records listed in day- and hour-specific files match records provided in the annual inventory inputs. This is necessary because SMOKE builds its list of all point sources based on the annual and/or average-day inventory files, and these are the only sources that will be listed in the SMOKE intermediate inventory file. **Smkinven** requires that the year associated with the date of the day-specific and hour-specific emissions is the same as the base year. **Smkinven** generates an error message when it cannot match a day- or hour-specific source with an inventory source. Please refer to Section 6.13, "**Smkinven**" [260] for more information about how **Smkinven** can be used to import these data.

Smkinven also can process the hourly CEM data in a more sophisticated way. Hourly heat input from the CEM data are used to allocate annual emissons to hourly emission data. First of all, the utility program **CEMScan** must be run before **Smkinven**, see more information at (Section 5.3.4, "**CEMScan**" [190]).

2.9.19. Processing hour-specific CEM data

Smkinven can allocate annual emissions to hourly data using the hourly heat input from the standard local time CEM data. **Smkinven** must match the ORIS/boiler combinations that appear in the standard local time CEM data to sources in the annual emissions inventory. **Smkinven** will first skip any ORIS/boiler combinations in the input CEM data that do not appear in the summary list (CEMSUM) created by CEMScan; it is important that users ensure that their input CEM data and CEM summary file remain consistent. **Smkinven** will also skip any CEM ORIS IDs that are not in the inventory and also any ORIS and boiler combinations that are not in the inventory. Note that the inventory may contain sources with valid ORIS IDs but blank boiler codes; none of these sources will be matched to the CEM data. The format of the CEM hour-specific data is shown in Table 8.22, "CEM Format for individual hour-specific data files" [370].

Data Check : Before calculating the hourly emissions, **Smkinven** first checks if the hourly NOx emissions from the CEM data are zero or null. If so, **Smkinven** checks if the hourly heat input, gross load, and steam load values are zero or null before all output values will be set to zero. In this case a warning will be written to the **Smkinven** log file.

Special handling is needed when processing SO_2 and NO_x emissions. If the summed annual CEM SO_2 and NO_x emissions for a particular ORIS/boiler combination are zero or null, **Smkinven** will calculate hourly emissions based on the SO_2 and NO_x emissions in the annual inventory rather than using the hourly CEM data. If the summed annual SO_2 and NO_x emissions are valid but a particular hour is missing the hourly SO_2 and NO_x emissions, **Smkinven** will set the hourly SO_2 and NO_x emissions to zero and write a warning to the log file.

Calculation: When calculating hourly emissions, **Smkinven** must use data from the annual inventory. If any of the matching inventory sources are missing the annual emissions value, the program will exit with an error. **Smkinven** will start with the hourly emissns from the CEM data and then disaggregate the emissions to the matching inventory sources like so:

Hourly NOx emissions for source i *= (annual NOx emissions for source* i */ summed annual NOx emissions for all matching sources) * hourly CEM NOx emissions * pounds to tons conversion*

If the summed annual emissions for all matching sources is zero, the hourly CEM emissions will be distributed evenly to the matching inventory sources. For all other pollutants, the hourly emissions are calculated as;

Hourly emissions for source i *= annual factor * annual emissions for source* i

The annual factor in the above calculation will preferentially be:

Annual factor = hourly heat input for ORIS/boiler / annual summed heat input for ORIS/boiler

If heat input data are not available, **Smkinven** will fallback to steam load followed by gross load. The heat input values do not need to be disaggregated to the matching inventory sources because the same disaggregation factor would be used for both the hourly heat input and summed heat input.

Used to calculate hourly flow rates from hourly heat input when reading CEM data. If FLOW_RATE_FACTOR is set to zero or unset, then **Smkinven** will not calculate hourly flow rates.

*Hourly flow rate (m^3/s) = [FLOW_RATE_FACTOR (ft^3/MMBTU) * hourly heat input (MMBTU/hr) * 0.02831 m^3/ft^3] / 3600 s/hr*

Smkinven then needs to assign the ORIS/boiler-level hourly flow rate to the matching inventory sources. To do this, it sums the flow rate for sources with the same stack. **Smkinven** uses the plant ID and stack ID to determine which sources feed into the same stack. Script setting information is available at Section 6.13.3.2, "Input Environment Variables" [265].

NOTE: It is not recommended to run **Smkinven** for an entire year's worth of CEM data since the memory requirements are large. Instead, users should process only the CEM data for their episode of interest. This may require multiple runs of **Smkinven** to break up large episodes.

2.10. Temporal processing

The temporal allocation of emission inventory data always occurs after the inventory import processing previously described in Section 2.9, "Inventory import" [53]. The **Temporal** program processes data for anthropogenic sources, while the **Tmpbio** program allocates biogenic emissions. In this section, we focus on the temporal allocation of the anthropogenic inventories using **Temporal**. The biogenic processing is further described in Section 2.17, "Biogenic processing" [91].

The primary purpose of the **Temporal** program is to create an intermediate hourly emissions file (ATMP, MTMP, or PTMP). It also creates a supplementary intermediate file that indicates which monthly, weekly (day-of-week), and diurnal (hourly) profiles were assigned to each source (ATSUP, MTSUP, or PTSUP). Since the **Temporal** dynamically create names for the output files, two new environment variables [A|M|P]TMPNAME and [A|M|P]TSUPNAME are used to set the directory and file prefix for naming the output files [A|M|P]TMP and [A|M|P]TSUP. The files are named using the starting date of each time period.For example, if ATMPNAME is set to /data/ntmp.nctox., then the ATMP file for a given time period will be put in the data directory and named ntmp.nctox.[start date].ncf.

The temporal processing operation applies factors based on the source characteristics to the emissions data from the SMOKE inventory files. These factors can include monthly, weekly, and diurnal temporal profiles. The resulting emissions data vectors (not a matrix) contain hourly emissions for the inventory species. SMOKE assumes an hourly time step (Even though the time step is an input setting to SMOKE, it currently cannot be changed.). Most of the calculations are implemented as sparse-matrix algebra based upon temporal cross-references and profiles, augmented by the substitution of values from day- and hour-specific

emissions data sets. For mobile sources, hourly emissions values also depend on meteorology (e.g., the temperature dependence of evaporative emissions).

Figure 2.20, "Transformation of inventory data to hourly data" [64] shows how data from the intermediate inventory are stored in the hourly file. The arrow represents the temporal processing steps which convert the annual, average-day, or day- and hour-specific data to hourly data. After the temporal processing, the hourly emissions are stored in the intermediate hourly file, by hour and source number. The emissions are stored in the same order as the sources in the sorted intermediate inventory file.

Figure 2.20. Transformation of inventory data to hourly data

Intermediate inventory file

Src	Emissions
1	10.0
2	15.0
3	10.0
4	20.0
5	15.0
6	10.0
7	15.0
8	10.0
9	20.0

Intermediate hourly file

Hour	Emissions
1	1.0
1	1.5
1	0.7
1	3.0
1	2.0
1	0.5
1	1.8
1	1.1
1	2.7
2	1.2
2	1.6
2	0.9
2	3.3
2	2.2
2	0.6
2	2.0
2	1.3
2	2.8
3	0.1
3	0.4
3	0.2
3	0.9
...	...
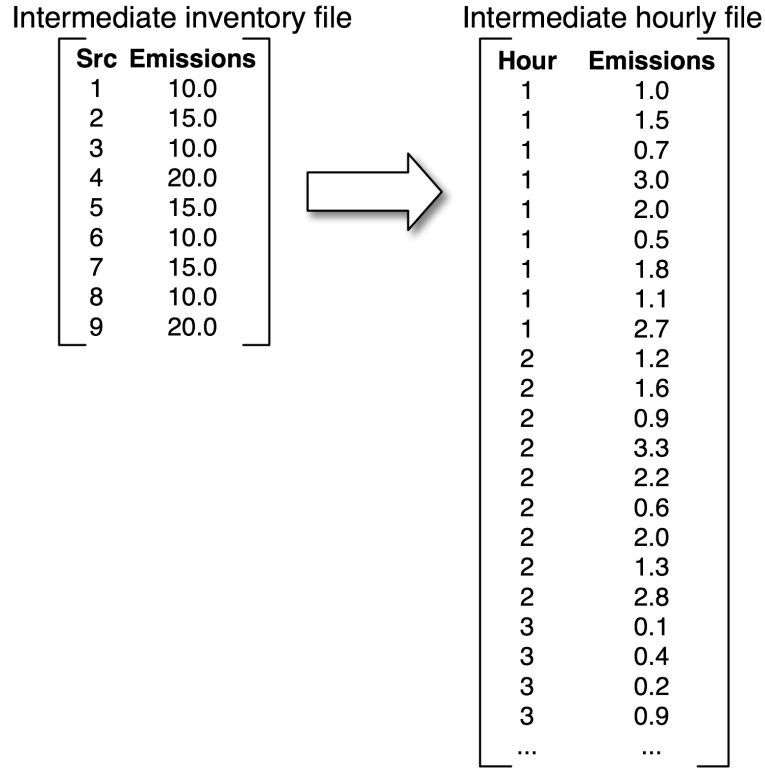
Temporal processing also addresses the following issues that need to be considered during emissions processing:

1. Using annual or average-day data when both are available in the inventory

2. Applying monthly, weekly, and diurnal profiles

3. Using day- and hour-specific emissions

4. Time zone adjustments

5. Holiday processing

6. Monday-weekday-Saturday-Sunday (MWSS) processing

7. Processing non-sequential dates

8. Creating the intermediate files

In the subsections below, we address each of these issues in the same order as the list above.

2.10.1. Using annual or average-day data

When using the ORL or FF10 inventory format, you may choose to use either the annual or the average-day emissions values when running the **Temporal** program. The default is to use the annual data. To apply average-day data instead, the `SMK_AVEDAY` setting is used (this setting is relevant only for ORL or FF10 inventories). These emissions are then used in the merge-processing step, resulting in model-ready emissions that depend on the data type selected. If part of your inventory is available as average-day data and part is available as annual data, you have used **Smkinven** to fill in annual values based on the average-day values. The **Temporal** program is then run using annual values. **Temporal** ensures that for those sources for which "annual" values were created, only day-of-week and hourly adjustments (not monthly profiles) are applied. **Temporal** assumes that any average-day data provided has already been adjusted for a specific month, and therefore does not apply the monthly profiles to them.

2.10.2. Applying monthly, weekly, and diurnal profiles

The **Temporal** program uses a cross-reference file and a temporal profiles file to assign the temporal profiles to the inventory sources. The cross-reference file can assign the profiles using a hierarchy that is based on the source characteristics. A detailed list of valid assignments is given in Section 6.16, "**Temporal**" [291] with the detailed description of the **Temporal** program. The cross-reference assigns a monthly, weekly, and diurnal profile to each source, and can also assign default profiles to sources that don't have more specific matches in the cross-reference file. When default assignments are made by the **Temporal** program, you can optionally choose to have warnings given for all of these assignments, using the `REPORT_DE-FAULTS` option.

The temporal profiles file has different sections that contain the monthly, weekly, and diurnal profiles. Additionally, you may provide different weekend (Saturday, Sunday) and weekday (Monday through Friday) diurnal profiles, or you may provide a different set of diurnal profiles for each day of the week.

When using multiple diurnal profiles, a given source must use a diurnal profile with the same profile code for each day. For example, if a source uses diurnal profile 1 for Monday, it will also use profile 1 for Tuesday, but Monday's profile 1 can be different than Tuesday's profile 1.

For weekly profiles, **Temporal** evaluates the TPFLAG setting (set by **Smkinven** as explained in Section 2.9.13, "Set the weekday averaging approach" [60]) and uses the weekly-normalized weekly profiles or weekday-normalized weekly profiles, depending on the setting.

One limitation of using monthly, weekly, and diurnal profiles is that there is no week-to-week variation within a month. For example, all Tuesdays in a month use the same emissions profile (unless one happens to be a holiday, see Section 2.10.5, "Holiday processing" [67]). This limitation can be overcome by using day-specific emissions data (only for point sources).

2.10.3. Using day- and hour-specific emissions

SMOKE uses the most specific data available for any given source. For point sources, **Temporal** will read in the day-specific and hour-specific emissions and use these instead of the annual or average-day emissions that have been adjusted to hourly values. If both day-specific and hour-specific data are available for the same source, **Temporal** will use the hour-specific data. The settings that control the use of these emissions are `HOUR_SPECIFIC_YN` and `DAY_SPECIFIC_YN`.

2.10.4. Time zone adjustments

You can use the `OUTZONE` setting to control the output time zone that defines the hours in the output files. The one constraint on this setting is that it must be consistent with the time zone of the meteorology files. These files are used only when processing on-road mobile sources with MOVES/MOBILE6 (optional), processing elevated point sources for the CMAQ-based approach (required), or processing biogenic emissions with SMOKE BEIS3 (required). The time zones must be consistent among all source categories processed, so if one source category depends on the meteorology file, then all source categories must be processed with the same `OUTZONE` setting.

Temporal compares the `OUTZONE` value with the time zone of the source, which was set in **Smkinven** (see Section 2.9.14, "Assign country codes, years, and time zones" [60]) based on the county. Additionally, it assesses and accounts for whether the date being processed falls within the range of Daylight Savings Time, and whether the county of the source uses Daylight Savings Time. **Temporal** uses the `COSTCY` or the `GEOCODE_LEVEL4` (if USE_EXP_GEOCODES Y) file to determine which counties use Daylight Savings Time and which do not; for example, the state of Arizona does not use it. Using these pieces of information, **Temporal** interprets the diurnal profiles assuming that they are local profiles in order to map the correct adjustment to the correct output hour in the output time zone. **Temporal** also uses the time zone of the source and the output time zone to determine the correct hour for switching from one month to the next and from one day of the week to the next.

Table 2.13, "Example `OUTZONE` settings and their associated time zones" [67] lists a sampling of `OUTZONE` settings and the time zones that they represent. Note that SMOKE expects `OUTZONE` to be set as a positive number for time zones in the Western Hemisphere, although standard notation would list these as negative values. For example, Eastern Standard Time is listed in this table as -5:00 hours from GMT, but `OUTZONE` for EST in SMOKE is 5. One result of this implementation is that SMOKE does not work perfectly for time zones east of GMT.

Table 2.13. Example `OUTZONE` settings and their associated time zones

OUTZONE	GMT	Time zone	Description
12	-12:00	BIT	Baker Island Time
11	-11:00	SST	Samoa Standard Time
10	-10:00	HST	Hawaii Standard Time
9	-9:00	AKT	Alaska Standard Time
8	-8:00	PST	Pacific Standard Time
7	-7:00	MST	Mountain Standard Time
6	-6:00	CST	Central Standard Time
5	-5:00	EST	Eastern Standard Time
4	-4:00	AST	Atlantic Standard Time
3	-3:00	ART	Argentina Time
2	-2:00	FNT	Fernando de Noronha Time
1	-1:00	EGT	Eastern Greenland Time
0	0:00	GMT	Greenwich Mean Time
-1	1:00	CET	Central European Time
-2	2:00	EET	Eastern European Time
-3	3:00	MSK	Moscow Time
-4	4:00	GST	Gulf Standard Time
-5	5:00	PKT	Pakistan Standard Time
-6	6:00	BST	Bangladesh Standard Time
-7	7:00	THA	Thailand Time
-8	8:00	HKT	China Standard Time
-9	9:00	KST	Korean Standard Time
-10	10:00	AET	Australian Eastern Time
-11	11:00	ADT	Australian Daylight Time
-12	12:00	FJT	Fiji Time
-13	13:00	NZT	New Zealand Daylight Time
-14	14:00	LNT	Line Islands Time

2.10.5. Holiday processing

Holidays that fall on weekdays can have different activity patterns than regular weekdays because commercial and commuting activities are altered. **Temporal** uses the HOLIDAYS file to identify which user-defined dates should be processed as holidays and to determine which day of the week (Saturday or Sunday) to use to model the holiday. Usually, holidays are modeled as if they were a Sunday; this attempts to account for things like plants being closed and traffic patterns being different. Of course, you can also set up specific temporal profile inputs for a specific holiday and model that day separately, but this is not automated in SMOKE. Users therefore simply pick Saturday or Sunday as an alternative date treatment in the hope that it will somewhat better represent the emissions than using a weekday.

Holidays processing happens automatically if you are running the **Temporal** program for every day of your episode. If you are using the Monday-weekday-Saturday-Sunday approach described in Section 2.10.6, "Monday, weekday, Saturday, Sunday processing" [68], then additional scripting steps must be taken to ensure that the holidays are modeled properly. In particular, for inventories that span multiple time zones and/or when the output time zone is not the same as at least one time zone in the inventory, the scripts must be configured to model the holiday and the next day, so that the final hours in the holiday that are west of the output time zone will be included in the next day's file.

2.10.6. Monday, weekday, Saturday, Sunday processing

The temporal variation of the days in a week is the same from week to week within a month because SMOKE uses weekly profiles. Also, it is common that the weekly profiles do not vary on Monday through Friday. Consequently, it is often desirable for long (e.g., annual) simulations to use a Monday-weekday-Saturday-Sunday (MWSS) approach. With this approach, SMOKE computes emissions for a representative Monday, weekday, Saturday, and Sunday within each month. The representative days cannot be the first day of the month (to prevent effects from the previous month from being included in the emissions data), a holiday (as set in the HOLIDAYS file), or the day after the holiday. Monday is distinguished from other weekdays because in multi-time zone cases in the Western Hemisphere with OUTZONE set to 0 (i.e. GMT), a few late-night Sunday hours are included in the hours at the start of the Monday file. In addition, one must specifically process the holidays and the day after holidays as separate runs for all holidays set by the HOLIDAYS file. During merging of emissions, the Monday, weekday, Saturday, and Sunday files are reused to create model-ready emissions for every day and hour needed. The holiday and day-after files are used for the holiday and day-after dates only.

This approach relies heavily on scripting to select dates for each month's representative days and to ensure that the days that are run are consistent with the dates in the HOLIDAYS file. Scripting is also responsible for ensuring that the correct files are merged together by **Smkmerge** to create the model-ready files for all days of the episode. These scripting issues are described in more detail in Chapter 4, *Using SMOKE Scripts* [117].

There are some situations in which you cannot use the MWSS approach. In particular, on-road processing with MOVES must be run for all days of the episode, although some steps can be sped up by using the meteorology averaging approach. Additionally, biogenic emissions processing obtains all of its temporal variation from the meteorology data, and therefore must be run for all days. Fortunately, biogenic emissions processing is very fast compared to processing for other source categories.

2.10.7. Processing Non-sequential Dates

Previously, **Temporal** processed a single continuous time period during each execution of the program producing one output file. Typically, only representative Monday, weekday, Saturday and Sunday, plus any holidays are processed for a single month. This type of processing can require complex scripting and **Temporal** would need to be run several times. See Section 4.4.13.3, "Changing the episode" [162], Section 4.4.13.4, "Changing the dates, times, and duration of model-ready SMOKE emissions files" [163] and Section 4.4.13.5, "Setting non-sequential processing dates" [164].

SMOKE now provides the capability for optionally setting non-sequential date processing during a single execution of **Temporal** using a new input file PROCDATES to indicate a list of dates which **Temporal** should process. The format for PROCDATES is described in Section 8.3.4, "PROCDATES: Procdates date list" [386]. The format allows for blank lines, comment lines (any lines that start with a pound sign, and trailing comments (any characters after an exclamation point).

G_TSTEP will be used to set the time step for all time periods. The output data for each time period will be written to an endividual file.

An example of `PROCDATES` file is shown below. In this case, we are processing the first day of each month in 2005. Twelve output files will be produced. Each file will contain 25 time steps.

- # First day of each month

- 20050101 0 250000 ! January

- 20050201 0 250000 ! February

- 20050301 0 250000 ! March

- 20050401 0 250000 ! April

- 20050501 0 250000 ! May

- 20050601 0 250000 ! June

- 20050701 0 250000 ! July

- 20050801 0 250000 ! August

- 20050901 0 250000 ! September

- 20051001 0 250000 ! October

- 20051101 0 250000 ! November

- 20051201 0 250000 ! December

Temporal procduces two output files `[A|M|P]TMP` and `[A|M|P]TSUP`. Since **Temporal** will need to dynamically create names for the output files, we will use two new environment variables `[A|M|P]TMPNAME` and `[A|M|P]TSUPNAME` to set the directory and file prefix used to name the output files. The files will be named using the starting date of each time period. For example, if `ATMPNAME` is set to /data/ntmp.nctox., (note the period at the end of the file string) the `ATMP` file for a given time period will be put in the `STATIC` directory and named "ntmp.nctox.<start date>.ncf".

2.10.8. Creating the intermediate files

Finally, the **Temporal** program must output intermediate files. The hourly emissions are written to the `ATMP`, `MTMP`, and `PTMP` I/O API files. Unlike the other major SMOKE intermediate files (e.g., the matrices), the actual emissions (not just factors) are written to this file. This is because day-specific and hour-specific emissions can be impossible to convert into factors since the annual inventory emissions for the day- or hour-specific sources could be zero and factors would not be able to change that.

If **Temporal** has more than 120 variables to output (the limit for the number of variables in an I/O API file), **Temporal** opens as many files (using the FileSetAPI wrapper) as are needed to store the data. SMOKE also estimates how large the output files will be using 120 variables per file and automatically lowers the number of variables that will be put in each I/O API output file to ensure that the files use less than 2 GB of disk space. In addition, **Temporal** writes the supplementary files `ATSUP`, `MTSUP`, or `PTSUP`, which contain the temporal profiles assigned to each source. The structures of the SMOKE intermediate files output by **Temporal** are provided in Section 9.14, "**Temporal**" [480].

2.11. Chemical speciation processing

An emission inventory is built and reported for a variety of pollutants, such as CO, NO_x, VOC, PM_{10}, and SO_2. However, AQM chemical mechanisms (e.g., CB4) contain a simplified set of equations that use "model species" to represent atmospheric chemistry. Therefore, emissions processing requires speciation profiles to convert the emissions in terms of pollutant values to the species used in the photochemical mechanism. The purpose of the chemical speciation processing program **Spcmat** is to produce matrices that contain the factors for converting the input emissions pollutants to the model species used in the AQM. These species include organics, PM species, and toxics species.

The speciation matrices that **Spcmat** creates are used transform column vectors of inventory-pollutant emissions into column vectors of model-species emissions. As shown in Figure 2.21, "Relationship of inventory sources to speciation matrix" [70], the speciation matrix consists of columns for each required pollutant-to-species transformation and includes an entry for each source. The entries are the factors needed to convert the inventory pollutants into the model species. Note that speciation matrices depend only upon the chemical mechanism and the inventory, and they are therefore independent of the other factor-based operations for emissions processing.

Figure 2.21. Relationship of inventory sources to speciation matrix

Intermediate inventory file

Src	Emissions
1	10.0
2	15.0
3	10.0
4	20.0
5	15.0
6	10.0
7	15.0
8	10.0
9	20.0

Speciation matrix

CO to CO	NOX to NO	NOX to NO2	VOC to ALD2	...
1.0	0.9	0.1	0.04	...
1.0	0.9	0.1	0.05	...
1.0	0.9	0.1	0.05	...
1.0	0.9	0.1	0.04	...
1.0	0.9	0.1	0.04	...
1.0	0.9	0.1	0.04	...
1.0	0.9	0.1	0.05	...
1.0	0.9	0.1	0.04	...
1.0	0.9	0.1	0.05	...
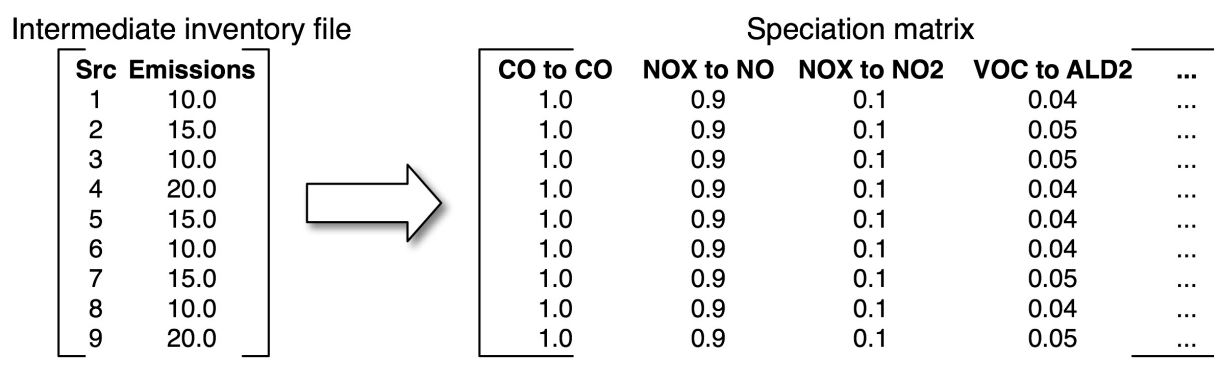
Chemical speciation processing addresses the following issues during emissions processing:

1. Splitting inventory pollutants into chemical species

2. Pollutant-to-pollutant conversions

3. Checking the consistency of the speciation profiles with the inventory

4. Setting the order of the output species

5. Creating speciation intermediate files

In the subsections below, we provide additional detail about each of these steps, in the order in which they are listed above.

2.11.1. Splitting inventory pollutants into chemical species

SMOKE supports run-time, user-selected inventory pollutants and chemical mechanisms. Before running the chemical speciation step, the only relevant information that SMOKE has is information about the inventory pollutants. After **Spcmat** runs, SMOKE then has the instructions for supporting a specific chemical mechanism, and through the speciation matrices, SMOKE will be able to generate model-ready emissions

for the specific chemical mechanism set by the user. The inventory pollutants relate to the chemical mechanism because certain pollutants are needed to create certain species, but the pollutants do not dictate the chemical mechanism. As a SMOKE user, you must be aware of what pollutants are required to generate the model species needed by a chemical mechanism, so that all needed model species are created.

As we just mentioned, SMOKE learns of the species being created for a given run through the **Spcmat** program. The chemical speciation profiles input file (GSPRO) is the data file that controls the chemical species SMOKE will create. It contains the chemical speciation profile code, the pollutant-to-species relationships, and both mole-based and mass-based conversion factors. The format and contents of this file are described in detail in Section 8.5.2, "GSPRO: Speciation profile file" [390]. Several GSPRO files to support many chemical mechanisms are available for download; see Section 4.2, "Test case descriptions" [117] for more information. How to select an existing chemical mechanism or specify a new one, and how to ensure that the GSPRO file is consistent with other input files, are described in Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150].

Spcmat uses a cross-reference file to assign the chemical speciation profiles to the inventory sources and pollutants. The cross-reference file can assign the profiles using a hierarchy that is based on the source characteristics; a detailed list of valid assignments is given in Section 6.15, "**Spcmat**" [286]. All assignments are pollutant-specific, such that each pollutant for a source can (and often should) use a different speciation profile. The cross-reference file can also assign a default profile, and some pollutants that have only one way of being speciated (e.g., mapping the CO pollutant to the CO species) receive a default profile for every source. When **Spcmat** makes default assignments, you can optionally choose to have warnings given for all of these assignments using the REPORT_DEFAULTS option. **Spcmat** will also produce a warning about any inventory pollutants that are not assigned a speciation profile, because this will result in the emissions for that pollutant being dropped by SMOKE.

Spcmat creates two speciation matrices during each run: a mole-based matrix and a mass-based matrix. The speciation profile file (GSPRO) has different factors for mass- and mole-based conversions. It is not trivial to convert between mass- and mole-based factors for some chemical mechanisms like CB4, which use aggregates of chemical compounds or parts of compounds to define the model species. One cannot simply use a molecular weight to convert accurately, because the molecular weight of the chemical species is different for every speciation profile. This is because different proportions of chemical compounds are present in each speciation profile, so even though the species are the same, their molecular weights are different from profile to profile. This is why SMOKE has the two speciation matrices. The mole-based matrices are used to create the model-ready files, and the mass-based matrices are used only to create the reports that require tons, kilograms, grams, or other mass units. One peculiarity of the mole-based matrix is that particulate species emissions cannot be expressed in moles, so the units are still grams in the mole-based matrices for particulate species.

Chemical speciation has both similarities and differences from the aggregation and disaggregation that is performed during inventory import (see Section 2.9.7, "Aggregate or disaggregate toxics emissions" [57]). It is similar in that it involves separating one data value into more than one data value. For example, inventory disaggregation can split Mercury & Compounds (CAS=199) into elemental mercury, divalent gaseous mercury, and divalent particulate mercury during inventory import of nonroad mobile sources. Similarly, NO_x can be split into nitrogen oxide (NO) and nitrogen dioxide (NO_2) during chemical speciation. Aggregation is also similar to speciation because it can map multiple pollutants or parts of pollutants into the same chemical compound, just as speciation can map parts of the different pollutants (e.g., HGSUM and HG) to the same model species (e.g., HG). On the other hand, the two concepts are different for three reasons. First, the pollutants aggregated and disaggregated by **Smkinven** are still considered by SMOKE to be pollutants, not species, to pass to later processing steps. Second, chemical speciation allows multiple split factors to be used for the same pollutant-to-species conversion, whereas inventory aggregation/disaggregation uses just one factor across the whole inventory for each pollutant-to-pollutant conversion. Third, chemical speciation also converts the units of the pollutants (tons) to the units of the species (moles for gaseous species and grams for particulate species).

Speciation profiles do not necessarily conserve mass. For example, it is possible to input 100 tons of VOC into the **Spcmat** program and have it output factors that will produce 70 tons of VOC-based species or 110 tons of VOC-based species. The reduced mass occurs when some of the pollutant's mass does not map to chemically reactive species in the inventory. In some cases, the nonreactive (NR) species is included in the speciation profiles so that the speciation profiles do sum to 1. Increased mass happens because some compounds that are part of VOC may have chemical reactivity associated with two model species. Since this one part of the VOC is mapped to two model species, its mass appears to be double-counted when summing the model-species mass. This is merely an artifact of how the model species are defined and implemented in the AQM, and the AQM is responsible for accounting for such issues in its chemical mechanism.

2.11.2. Pollutant-to-pollutant conversions

In some cases, the pollutant available in the inventory is not the same as the pollutant for which speciation profiles have been developed. At this time, the only known case of this situation is when the inventory is collected for VOC (or for reactive organic gases [ROG]) but the speciation profiles are available for TOG. For this situation, VOC-to-TOG factors have been developed that **Spcmat** assigns to the inventory sources based on the sources' SCCs. The factors are input using the GSCNV file, which can support multiple pollutant-to-pollutant conversions in the same file. The example SMOKE GSCNV file has sections for ROG-to-TOG and VOC-to-TOG; it is described further in Section 4.2, "Test case descriptions" [117] and Section 8.5.1, "GSCNV: Pollutant-to-pollutant conversion file" [389].

2.11.3. Checking the consistency of the speciation profiles with the inventory table

The **Spcmat** program checks to be sure that the two definitions of the NONHAPVOC pollutant - one from the speciation profiles (GSPRO) file and the other from the inventory table (INVTABLE) file - are consistent. When you are modeling with both criteria and toxics inventories and using toxics VOC (i.e., "integrating" toxics VOC), you must ensure that the definition of toxics VOC used in the INVTABLE file is consistent with the definition in the GSPRO file. **Spcmat** cannot assure that these files are consistent with each other, but does ensure that the information provided by the user is consistent; it is up to you to make sure that the information provided is correct and consistent with what is actually in the files.

These files need to be consistent because the same pollutants that SMOKE subtracts out of the VOC to create NONHAPVOC must also be removed when the speciation profiles are being computed. This assures that there will be no double-counting of mass when calculating model species from both toxics VOC and from the criteria aggregated VOC value. The GSPRO file format includes a header that allows you to define the toxics VOC that were removed when creating the NONHAPVOC speciation profiles. This is intended to help you ensure that the NONHAPVOC definition in the GSPRO file is consistent with the inventory table file. One important detail of how this is implemented is that **Spcmat** will not include inventory pollutants that are not actually in the inventory in the consistency check, even if the pollutant is listed in the NONHAPVOC definition in the GSPRO file.

2.11.4. Setting the order of the output species

The **Spcmat** program controls the order of the species in the output files. The order does not matter to any of the AQMs that SMOKE supports, so there is not very much user control of the order. **Spcmat** arranges the output species first by the order in which their associated pollutants appear in the inventory table (INVTABLE) file, and next in alphabetical order. For example, if an inventory included CO, NO_x, and VOC, the CO species from the CO pollutant would be first, followed by the NO and NO_2 species for NO_x. After that, all of the VOC species would be output in alphabetical order (e.g., ALD2, ETH, FORM, ISOP, NR,

OLE, PAR, TOL, XYL). If two pollutants create the same species, the first pollutant in the `INVTABLE` file with which the species is associated will determine its output order.

2.11.5. Creating speciation intermediate files

The last task for the chemical speciation processing is creating the speciation intermediate files. The **Spcmat** program creates a mole-based speciation matrix (`ASMAT_L`, `MSMAT_L`, or `PSMAT_L`) and a mass-based speciation matrix (`ASMAT_S`, `MSMAT_S`, or `PSMAT_S`), as previously mentioned. These matrices can have any number of pollutant-to-species conversions; the **Spcmat** program will open multiple speciation matrix output files if the number of pollutant-to-species conversions is greater than 120 (the limit for the number of variables in an I/O API file). **Spcmat** opens and writes as many files with 120 variables (using the FileSetAPI wrapper) as are needed to store the data. In addition, the **Spcmat** program writes a supplementary speciation file (`ASSUP`, `MSSUP`, or `PSSUP`) that contains the speciation profile assignments for each source. The structures of the SMOKE intermediate files output by **Spcmat** are provided in Section 9.13, "**Spcmat**" [479].

2.12. Spatial processing

The spatial processing operation, or *gridding*, combines the grid specification for the air-quality modeling domain with source locations from the SMOKE inventory file. The resulting gridding matrix is a sparse matrix that describes in which grid cells the emissions for each source occur within the modeling domain. The gridding matrix is applied to the inventory emissions to transform source-based inventory emissions to gridded emissions.

The SMOKE **Grdmat** program creates the gridding matrix for area, mobile, and point sources. The gridding step is different depending on the type of source being processed.

- For area sources, county-total emissions are spread among the cells intersecting the county through the use of gridding surrogates.

- For mobile sources, the data can be provided by county (as area sources are), or the data can be provided as line sources ("links"). County-based mobile emissions are apportioned with gridding surrogates, preferably with surrogates based on the different road types for which the mobile emissions are provided. The line-source emissions are apportioned depending on the length of the link in each cell.

- For point sources, emissions are apportioned to the grid cell intersecting the point.

As shown in Figure 2.22, "Relationship between inventory and gridding matrix" [74], the gridding matrix contains the SMOKE source IDs that intersect each grid cell, and the source-to-cell factors for each. The gridding matrix is a sparse matrix because each source intersects only a small number of cells relative to the total number of cells in the domain, and the storage format shown in the figure reflects a sparse storage format. In the example in Figure 2.22, "Relationship between inventory and gridding matrix" [74], source 1 intersects cells 1 and 2, with 10% of the emissions in cell 1 and 30% of the emissions in cell 2. The remainder of the source's emissions (60%) are either outside the grid or in other cells not shown in this example. Source 2 is completely inside the domain; 40% of its emissions are in cell 1, 30% are in cell 2, and 30% are in cell 4. If this were a point-source speciation matrix, there would be only one cell associated with each source because each point source exists in only one cell.

Figure 2.22. Relationship between inventory and gridding matrix

Intermediate inventory file | Gridding matrix

Src	Emissions
1	10.0
2	15.0
3	10.0
4	20.0
5	15.0
6	10.0
7	15.0
8	10.0
9	20.0

Cell ID	Src	Factor
1	1	0.1
1	2	0.4
1	3	0.2
2	4	0.1
2	1	0.3
3	2	0.3
4	6	0.5
4	2	0.3
4	7	0.3
...	...	...
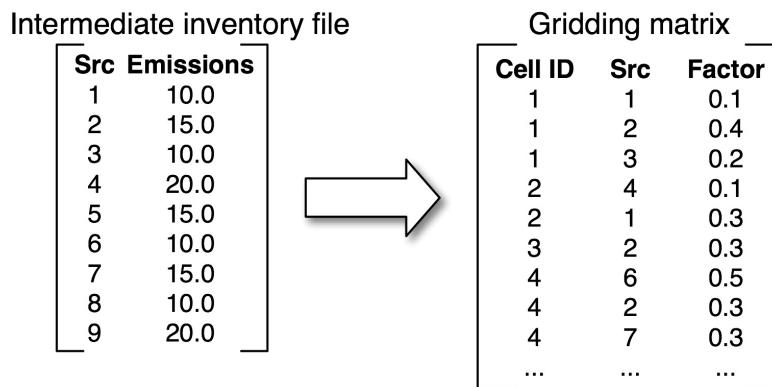
Note that the gridding matrix depends only upon the source locations, the grid definition, and in some cases gridding surrogates and cross-references. It is therefore independent of the other steps of emissions processing.

Spatial processing addresses the following issues during emissions processing:

1. Defining the gridded region to output from SMOKE

2. Assigning sources to grid cells

3. Creating the spatial allocation intermediate files

2.12.1. Defining the gridded region to output from SMOKE

The grid for which the spatial allocation step outputs the gridding matrix depends upon the `IOAPI_GRIDNAME_1` environment variable setting and the `GRIDDESC` input file. The file contains all of the settings needed by the I/O API to define each grid (together, these settings are called a grid definition), and it contains many such grid definitions. The `IOAPI_GRIDNAME_1` setting selects the specific grid to use during a specific run, and it must match the grid name provided in the `GRIDDESC` file. The settings used by the I/O API to define a grid are documented as part of the I/O API web site [http://www.baronams.com/products/ioapi/GRIDDESC.html].

There is another file format, called the `G_GRIDPATH` file, that can be provided to **Grdmat** instead of the `GRIDDESC` file. However, the `G_GRIDPATH` format allows only one grid per file. It is included as an acceptable input to SMOKE to allow SMOKE to be backwardly compatible with previous versions. All new users are advised to use a single `GRIDDESC` file to store all of their modeling grids, instead of using multiple `G_GRIDDPATH` files, one for each grid. The `IOAPI_GRIDNAME_1` setting is not used by **Grdmat** when the `G_GRIDPATH` format is used.

The gridded region selected at run time does not need to cover all of the counties in the inventory. If the gridded region is smaller than the inventory, this processing step will still include the counties or parts of counties that do not overlap the grid. This allows users who are performing emissions processing using nested grids to import the inventory once and apply gridding matrices for each grid to the same inventory, creating gridded emissions for all nested grids without having to adjust the inventory files. The downside of this approach is that SMOKE does not give a warning if a county that is in the inventory is not in the spatial surrogates, or if a lat-lon coordinate is not inside the grid. It assumes that these sources are intended to be dropped and proceeds without comment. Thus, you must ensure that your surrogate files contain the

counties in your inventory that are inside the grid and that your lat-lon coordinates for point and link sources are correct.

2.12.2. Assigning sources to grid cells

SMOKE takes a different approach to assigning sources to grid cells for each SMOKE source category and some special cases. In the following subsections, we describe the concepts of spatial allocation of (1) area/nonpoint, nonroad, mobile, and on-road mobile nonlink sources; (2) area-to-point sources; (3) pregridded area sources, (4) on-road mobile link sources, and (5) point sources.

2.12.2.1. Spatial allocation of area/nonpoint, nonroad mobile, and on-road mobile nonlink sources

The area/nonpoint, nonroad mobile, and on-road mobile nonlink sources all have emissions in the inventory with county-total values. To spatially allocate these emissions, factors must be assigned to each source to distribute the county-total emissions across the grid cells that intersect the county. This is accomplished using a cross-referencing approach that assigns a spatial surrogate to each source in the inventory.

The spatial surrogates files, located in SRGPRO_PATH contain factors for allocating emissions from a county-total to a gridded value, and there are usually many sets of factors available for each county. The data in this file are used to estimate the spatial distribution of county-total emissions inside the county. These sets of factors are calculated from other data that are available at a finer resolution than the county data, such as census tracks. Examples of commonly available surrogates are population, housing, urban area, rural area, agriculture, water, railroads, major highways, airports, ports, and forest. To ensure correct emissions processing, it is essential that all counties within the inventory and domain be included in the spatial surrogates file. If any counties inside the domain are left out of the file, then SMOKE will not be able to detect this; instead, it will act as though the county is outside of the domain and drop the emissions.

The general case of cross-referencing and profiles was described in Section 2.4, "Cross-referencing and profiles" [17]. During spatial allocation with the **Grdmat** program, the spatial cross-reference file (AGREF or MGREF) assigns spatial surrogates for area/nonpoint, nonroad mobile, and on-road mobile nonlink sources. The spatial cross-reference file associates SCCs with a spatial surrogate code, which is an arbitrary positive integer code that also appears in the spatial surrogates file along with the spatial surrogate data. It is important to ensure (if possible) that the spatial surrogate codes assigned to each SCC are actually available in the surrogate file for all counties in the inventory with that SCC. For example, if the water surrogate is assigned to motorboat sources in a given county, the spatial surrogate file should have values for the water surrogate in that county. If this is not the case, SMOKE will be forced to use a "fallback" surrogate, defined with the SMK_DEFAULT_SRGID environment variable setting, which assigns a surrogate that is defined for every county in the domain. SMK_DEFAULT_SRGID is mandatory and must be set to the population surrogate code listed in the SRGDESC file. The population surrogate in this case serves two purposes; 1) It defines all the FIPS codes contained within the gridded domain which reduces I/O in the gridding process and 2) this fallback approach prevents the emissions for that county and source category from being dropped from the emissions processing. **Grdmat** produces warnings in the log file whenever the fallback surrogate is used. If the fallback surrogate also causes the emissions to go to zero, an additional warning that indicates the emissions are being dropped is also written to the **Grdmat** log file. It is to the the users discretion to use the fallback described in purpose 2. This is set by the SMK_USE_FALLBACK environment variable either set as [Y/N].

The surrogate codes are defined, described and a reference to the spatial surrogate files listed in the surrogate description file SRGDESC which is likewise set as the environment variable SRGDESC. Because the surrogate codes are arbitrarily assigned, you need to make sure that the spatial cross-reference is developed in conjunction with the spatial surrogates so that the surrogate codes used in each are consistent. It is not generally wise to develop surrogates without a cross-reference, unless the surrogate codes used in the sur-

rogates file are set to the same values that are in a spatial cross-reference file that is already available. In other words, the surrogate codes in `SRGDESC`, `AMGREF` and `AMGPRO` should be in aggrement.

Figure 2.23, "Spatial allocation of county-total emissions" [76] illustrates spatial allocation of all county-based emissions sources. The box at the left represents a grid with a single county, shaded in gray to represent an emissions value in that county. The arrow represents the spatial allocation steps. The box at right shows that all grid cells that intersect the county have emissions contributed by the county. Cells that intersect more of the county are shown in darker colors, simulating the effect of using area surrogates. The darker cells could also represent cells in which a large city causes the population surrogate to concentrate the emissions in those cells, while the surrounding regions within the county have lower emissions.

Figure 2.23. Spatial allocation of county-total emissions
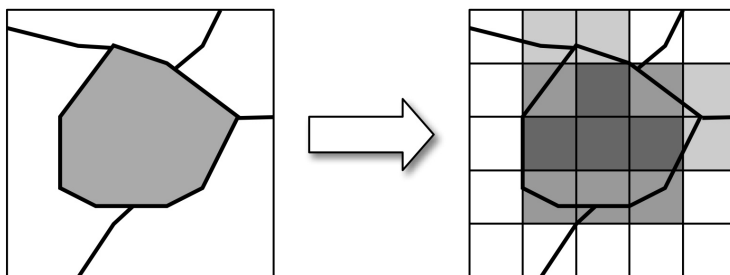
2.12.2.2. Spatial allocation of area-to-point sources

In some cases, the inventory import processing (by the **Smkinven** program) will have assigned point locations to some of the county-based inventory sources such as airports. Spatial allocation for these sources ignores the assigned surrogate, sets the surrogate assignment to 0, and uses the point-source locations and assigned fractions to determine the cells and associated magnitudes of the emissions from the county-total source.

2.12.2.3. Spatial allocation of pregridded sources

When pregridded data are provided to SMOKE, the spatial allocation step must still be run. In this case, **Grdmat** just maps the pregridded emissions to the correct cells and outputs a simple gridding matrix that is used in later processing steps to maintain the pregridded nature of the inventory.

2.12.2.4. Spatial allocation of on-road mobile link sources

On-road mobile link sources are straight line segments within a county that have emissions and/or VMT data associated with them as well as the latitudes and longitudes of the starting and ending positions of the link. SMOKE determines the fraction of each link within each grid cell, and then assigns the emissions or VMT from the link to those grid cells that the link intersects, by weighting the emissions according to the length of the link within a grid cell divided by the total length of the link.

As shown in Figure 2.24, "Spatial allocation of on-road mobile link sources" [77], three link sources in the left-hand diagram are joined at their ends to represent a road cutting through the county. As shown in the figure, link sources do not bisect county boundaries (new link sources would be used to continue the road into an adjoining county). The three portions of the link have different VMT values, as represented by the three different shades of gray that highlight the link. The arrow represents the spatial allocation step. In the right-hand diagram, the cells that intersect the link sources are shaded, with darker shading used in cells in which the link intersects the cell more and the link has higher VMT.

Figure 2.24. Spatial allocation of on-road mobile link sources
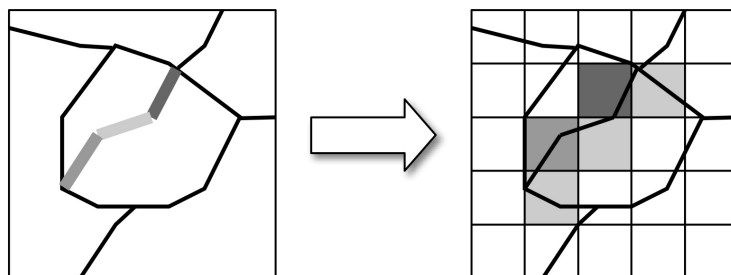
2.12.2.5. Spatial allocation of point sources

Spatial allocation of point sources in **Grdmat** is very straightforward. The **Grdmat** program simply determines which cell the lat-lon coordinates are in and assigns all emissions from that source to a single grid cell. Except for (1) plume-in-grid sources for all models and (2) elevated sources for UAM-based models (both addressed in Section 2.15, "Elevated-source processing" [83]), this means that the AQMs do not have information about where in the grid cell the point source is located.

2.12.3. Creating the spatial allocation intermediate files

The last task for the spatial allocation processing is creating the gridding intermediate files. The **Grdmat** program creates a gridding matrix (AGMAT, MGMAT, or PGMAT) and for on-road mobile sources an ungridding matrix (MUMAT). These matrices contain all of the factors needed to calculate gridded emissions from a source-based inventory. In addition, the **Grdmat** program writes a supplementary output file (AGSUP or MGSUP) that contains the spatial surrogates assigned for each source, indicates whether the assignment was a primary or a fallback, and identifies area-to-point sources. The structures of the SMOKE intermediate files output by **Grdmat** are provided in Section 9.4, "**Grdmat**" [465].

2.13. Growth processing

Growth processing creates emission data sets for years other than a year for which an emissions inventory is available. For example, if an inventory is available for 1996, but the modeling effort involves predicting ozone levels in 2007, then the emissions inventory must be grown to the year 2007. Previous versions of this document used the term "projection" for this function; in this version, however, "projection" refers to both growth of emissions (which is covered in this subsection) and control of emissions (addressed in Section 2.14, "Control processing" [80]). The **Cntlmat** program performs both growth and control functions. For growth processing, **Cntlmat** creates a growth matrix that contains the growth factors for each source and pollutant in the inventory. The **Grwinven** program then combines the growth matrix with the emission inventory to create a grown emission inventory.

If no new sources are being added when moving from the inventory year to the future year, then **Grwinven** can be used with the base case inventory and the growth matrix based on it. If new sources must be added, then the data structuring step (performed by **Smkinven**) must be rerun for the new number of sources, followed by running **Cntlmat** to create the growth matrix; then **Grwinven** applies the matrix to the new inventory file. Alternatively, users may elect to prepare a future-year inventory outside of SMOKE and import it directly with **Smkinven**, which skips the **Cntlmat** and **Grwinven** steps.

Figure 2.25, "Relationship between inventory sources and growth matrix" [78] shows the relationship between the inventory and the growth matrix created by **Cntlmat**, which consists of columns for each pollutant being grown from one year to another. The entries in the matrix are the growth factors needed

to grow the inventory to a future or past year; note that these entries can be greater than, equal to, or less than 1 depending on if the emissions should increase, stay the same, or decrease after the inventory is grown. If the growth factors are the same for every pollutant in the inventory, then only one column, called "pfac", is included in the growth matrix, rather than using duplicate columns for every pollutant. The growth matrix depends only upon the growth factors and the inventory, so it is therefore independent of other factor-based operations for emissions processing; however, growth of the inventory (using the **Grwinven** program) must occur before the temporal allocation step when creating model-ready emissions using inventories grown with SMOKE.

Figure 2.25. Relationship between inventory sources and growth matrix

Intermediate inventory file

Src	Emissions
1	10.0
2	15.0
3	10.0
4	20.0
5	15.0
6	10.0
7	15.0
8	10.0
9	20.0

Growth matrix

CO	NOX	VOC	PM2.5	...
1.1	0.8	2.2	1.0	...
1.2	0.7	1.9	1.0	...
1.2	0.5	1.8	1.0	...
1.1	0.8	1.5	1.0	...
1.4	0.9	2.0	1.0	...
1.3	1.0	1.4	1.0	...
1.3	0.7	1.6	1.0	...
1.5	0.8	2.1	1.0	...
1.0	0.6	1.8	1.0	...
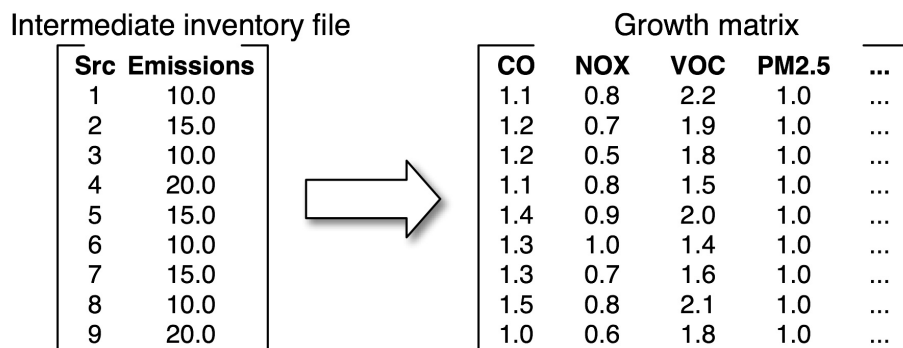
Grwinven combines the intermediate inventory files with one or more growth matrices to create a new intermediate inventory file with the same structure as the original file but with a future (or past) year stored in the header of this file.

In growth processing, the **Cntlmat** program addresses the following emissions processing needs when creating the growth matrix:

1. Assigning growth factors

2. Reporting on the factors assigned to each source in the inventory

3. Creating the growth matrix

The **Grwinven** program addresses the remaining needs to create a grown inventory:

4. Applying the growth matrix

5. Creating a grown inventory file

Each of the issues in the list above is addressed in the following subsections, in the order in which they appear in the list.

2.13.1. Assigning growth factors

The **Cntlmat** program assigns growth factors using a cross-reference approach similar to the approaches used for chemical speciation and gridding. **Cntlmat** reads the information about the growth factors from the /PROJECTION/ packet in the control input file (GCNTL). As described in detail in Section 6.2, "**Cntlmat**" [213], GCNTL can assign growth factors by state/county FIPS code, SCC, SIC, MACT, pollutant, and various combinations of these. The most specific entry is selected by **Cntlmat** based on the hierarchy described in Section 6.2, "**Cntlmat**" [213]. Since the /PROJECTION/ packet includes both the cross-referencing information and the growth factors, there is no need for a profile file (like those used for chemical

speciation). The growth factors may include both positive growth (factors greater than 1) or negative growth (factors less than 1) or no growth (factors equal to 1).

The /REACTIVITY/ packet can also be used to grow emissions to a future or past year, but a different approach is used that includes a "phase-in" period and other differences. We describe this packet in Section 2.14, "Control processing" [80].

2.13.2. Reporting on the factors assigned to each source in the inventory

In addition to actually applying the growth factors, the growth processing reports the factors that were applied to each source. **Cntlmat** writes a report that includes the source characteristics and the factor from the /PROJECTION/ packet assigned to the source. For example, the area-source report would include the state, county, and SCCs along with the factor assigned to the source. Only those sources that have growth factors applied are included in the report. This report permits you to verify that the assignments you intended were in fact made by the program. The report is called the APROJREP, MPROJREP, or PPROJREP file for SMOKE area, mobile, and point sources, respectively.

2.13.3. Creating the growth matrix

The growth matrix created by **Cntlmat** contains the growth factors for each source and pollutant. If pollutant-specific growth factors are used, the growth matrix contains one column for each pollutant that is grown. The names of the columns are the same as the names of the pollutants from the SMOKE inventory file, so that the **Grwinven** program knows which column to apply to which inventory pollutant. If the growth factors are applied uniformly to all pollutants in the inventory, then **Cntlmat** does not output a separate but identical factor for each pollutant. Instead, it writes only one column in the matrix, with the name "pfac".

2.13.4. Applying the growth matrix

The **Grwinven** program applies the growth factors to the inventory pollutants by simply multiplying the growth matrices for the pollutants with the annual and average-day emission values. It reads in the SMOKE intermediate inventory file and up to 80 growth matrices, then applies the growth factors from all growth matrices to the appropriate pollutants. If a growth matrix has the "pfac" variable, **Grwinven** applies the factors to all pollutants in the inventory; if instead the matrix has pollutant-specific entries, the factors are applied by matching the pollutants between the matrix and the inventory.

As is further described in Section 2.14, "Control processing" [80], **Grwinven** can also apply the control matrix at the same time it applies the growth matrix, and the total number of growth and control matrices that can be applied in one run is 80. The **Grwinven** program's growth processing features cannot be applied to a SMOKE intermediate file that **Grwinven** has created, because of the /FYEAR/ header element described in the next section.

2.13.5. Creating a grown inventory file

After applying the growth factors to the SMOKE intermediate inventory file, the **Grwinven** program writes a new SMOKE intermediate inventory file that contains the new, grown emissions. **Grwinven** only writes the I/O API part of the inventory and not the ASCII part, because the ASCII part that contains the state/county codes, SCCs, and other character strings does not change between the base and future or past year. All of the sources in the base and grown inventory files are the same; the difference is in their emissions values.

Grwinven adds a header element to the I/O API part of the inventory file that indicates the file was created using **Grwinven** and a growth matrix. The header element is /FYEAR/, which is followed by the date of the future year (for example, /FYEAR/ 2018). Other programs (such as **Temporal**) will recognize the /FYEAR/ header element and alter its messages to indicate that a future year is being processed.

2.14. Control processing

The control processing operation applies control factors from a control input file (`GCNTL`) based on source characteristics in the inventory. A control scenario involves changing the values of emissions based on regulations affecting industrial activities or personal behaviors. The resulting control matrix, created by the **Cntlmat** program, takes the form of the matrix shown in Figure 2.25, "Relationship between inventory sources and growth matrix" [78]. The control matrix depends only upon the source characteristics in the SMOKE inventory and the set of controls chosen, so control processing can therefore be decoupled from the rest of the processing steps. The **Cntlmat** program performs control processing for SMOKE area, mobile, and point sources; however, much more complex controls for on-road mobile sources can also be implemented when using MOVES through SMOKE to calculate emission factors and apply them to VMT.

The emissions control factors can be applied in addition to the emissions growth factors (described in Section 2.13, "Growth processing" [77]), and the net effect of this growth and control is called "projection". SMOKE control processing can create two types of control matrices during a given run: a multiplicative control matrix and a reactivity control matrix.

The **Cntlmat** program performs the following emissions processing steps in creating the control matrices:

1. Assigning control factors from six control packets to the sources

2. Creating the multiplicative control matrix

3. Creating the reactivity control matrix

4. Reporting on factors assigned to each source in the inventory

The **Grwinven** program addresses the following control processing steps:

5. Applying the multiplicative control matrices to the inventory

6. Creating a controlled intermediate inventory file

Finally, the **Smkmerge** program can be used to perform the following control processing step:

7. Applying the multiplicative and/or reactivity control matrices to the inventory to create model-ready inputs

The next seven subsections explain the concepts involved with these processing steps in more detail.

2.14.1. Assigning control factors from six control packets to the sources

SMOKE provides six control packets with which users can control emissions:

• /MACT/ contains MACT-based assignments for toxics inventories and can be used to apply general MACT controls to sources affected by MACT regulations. This packet contributes to the multiplicative control matrix.

- /CONTROL/ contains settings for control efficiency, rule effectiveness, and rule penetration that can be applied by nearly any combination of source characteristics, even targeting a specific source. This packet contributes to the multiplicative control matrix. This packet cannot appear in the same input file with an /EMS_CONTROL/ packet.

- /EMS_CONTROL/ contains settings for control efficiency, rule effectiveness, and rule penetration for both the base year *and* a future year. It also contains a point-source conversion factor and an aggregated control factor that can override everything else in the packet. This packet contributes to the multiplicative control matrix. This packet cannot appear in the same input file with a /CONTROL/ packet, and it can be used for point sources only.

- /CTG/ contains settings for control technology guideline (CTG) controls, MACT controls, and reasonably available control technology (RACT) controls. It contributes to the multiplicative control matrix.

- /ALLOWABLE/ contains county-specific, SIC-specific, SCC-specific controls, caps, and replacement emissions. It contributes to the multiplicative control matrix.

- /REACTIVITY/ contains settings needed for reactivity-based controls and its use results in the reactivity control matrix.

Section 6.2, "**Cntlmat**" [213] describes the **Cntlmat** program in more detail, including the cross-reference hierarchy of these packets and how they relate to one another. All packets can be included in a single **Cntlmat** run, with the exception of the /CONTROL/ and /EMS_CONTROL/ packets (either one or the other of these can be included, but not both). In general, these packets can assign control factors by state/county FIPS code, SCC, SIC, MACT, pollutant, other plant-specific source characteristics, and various combinations of these.

2.14.2. Creating the multiplicative control matrix

Cntlmat creates the multiplicative control matrix based on the information contained in the /MACT/, /CONTROL/, /EMS_CONTROL/, /CTG/, and /ALLOWABLE/ packets. The specific details of how these packets interact and how the control factors are calculated is described in the **Cntlmat** section of Section 6.2, "**Cntlmat**" [213]. Once the control factors have been calculated, **Cntlmat** writes a multiplicative control matrix file that contains one column of control factors for each pollutant; each row in the matrix represents a source. The names of the columns are the same as the names of the pollutants from the SMOKE inventory file, so that either the **Grwinven** or the **Smkmerge** program knows which control column to apply to which inventory pollutant.

2.14.3. Creating the reactivity control matrix

Reactivity controls have been included in SMOKE to allow users to examine what happens to the air quality modeling results when the chemical mixture of the emissions is changed to reduce its ozone-forming potential. Examining the effect of reactivity controls is known as "reactivity assessment". The implementation of this type of control includes permitting users to reset the base-year emissions, reset the SCC, and reset the chemical speciation profile. The controls can be applied to all sources that match an SCC or a specific facility or process within a facility.

Several issues are important when addressing emissions processing requirements for reactivity assessments. Reactivity assessment involves replacing one compound in the inventory by another compound. This replacement can impact emission projections, the total magnitude of the inventory pollutants, and the associated SCCs. The market penetration of the replacement compound may vary in time and space, which affects the future-year emissions. Also, the replacement compound may be needed in much greater or much smaller amounts, thereby affecting the total inventory emissions. Finally, if a different process is required in order for a source to use the different compound, the SCC for that source may change.

The scale of the reactivity assessment is important; it could be local, statewide, or national. A local case could involve investigating reactivity for one source. A statewide case could be implementing a change in compound based on reactivity considerations for a State Implementation Plan (SIP), and this would affect sources across the state. A national case could involve an EPA investigation of the formulation of nationally distributed consumer products.

In addition, exemptions from controls for certain sources must be permitted as part of an emissions control strategy. These exemptions can occur when a reactivity assessment determines that certain compounds and/or processes do not significantly affect pollution formation.

To address these issues, SMOKE is able to target changes in a VOC for specific classes of VOC emissions, and address the spatial and temporal considerations implied by market penetration issues. Furthermore, when replacement options are being investigated, the correct replacement operations are facilitated by SMOKE. These operations include selecting sources, changing underlying pollutant emissions, changing SCCs, correctly projecting future-year emissions based in part on market penetration issues, and appropriately speciating emissions for the new compound.

For a single run of **Cntlmat**, reactivity controls can be applied to only one pollutant, typically a VOC pollutant. Therefore, if you have more than one VOC pollutant in the inventory (e.g., a toxics VOC and a particulate pollutant), then separate reactivity matrices will need to be created for each pollutant that receives reactivity controls, and these controls will need to be applied in separate SMOKE runs (see Section 2.14.7, "Using **Smkmerge** to apply the multiplicative and/or reactivity control matrices" [83] for more information).

2.14.4. Reporting on factors assigned to each source in the inventory

In addition to assigning the control factors, **Cntlmat** also creates reports detailing the factors applied to each source. These reports include the source characteristics and the factors assigned to the sources. The reports are:

- `ACREP, MCREP, or PCREP:` For area and mobile sources, this report includes emissions before and after all multiplicative controls have been applied by state, county, and SCC. For point sources, the report includes emissions before and after all multiplicative controls by facility.

- `AREACREP, MREACREP, or PREACREP:` For pollutants controlled with reactivity controls (usually VOC), this report includes source number, base-year emissions, replacement base-year emissions, projection factor, future-year SCC, future-year speciation profile number, and market penetration rate of reactivity control. This report is the only reactivity report that can be generated by SMOKE since the **Smkreport** program cannot yet import and apply the reactivity matrix to generate reports.

- `ACSUMREP, MCSUMREP, or PCSUMREP:` For each source, this report includes the specific multiplicative controls that were applied to each source. It includes the emissions before and after each control, as well as the control factor that was applied. The control packet from which each control came is included in the report. The reactivity controls are not included in this report.

2.14.5. Using Grwinven to apply the multiplicative control matrices

The **Grwinven** program applies the multiplicative control matrices to the inventory pollutants by multiplying the control factors for the pollutants with the annual and average-day emission values. (Reactivity matrices cannot be applied by **Grwinven**; only the **Smkmerge** program can apply those, as discussed in Section 2.14.7, "Using **Smkmerge** to apply the multiplicative and/or reactivity control matrices" [83].)

Grwinven reads in the SMOKE intermediate inventory file and up to 80 control matrices and then applies the control factors from all matrices to the appropriate pollutants.

As discussed in Section 2.13.4, "Applying the growth matrix" [79], the **Grwinven** program can also apply growth matrices at the same time it applies control matrices, and the total number of growth and control matrices that can be applied in one run is 80. If desired, the control processing features of **Grwinven** can be used on a SMOKE intermediate file that has been created by **Grwinven** with grown and/or controlled emissions, as long as no *growth* processing features are included in that second pass. (See section Section 2.13.4, "Applying the growth matrix" [79] for discussion of this restriction on applying growth matrices, caused by **Grwinven**'s addition of the /FYEAR/ header element.)

2.14.6. Creating a controlled intermediate inventory file

After applying the control factors to the SMOKE intermediate inventory file, the **Grwinven** program writes a new SMOKE intermediate inventory file that contains the new, controlled emissions. **Grwinven** writes only the I/O API part of the inventory and not the ASCII part, because the ASCII part that contains the state/county codes, SCCs, and other character strings does not change between the base and future or past year. All of the sources in the base and controlled inventory files are the same; the difference is in their emissions values.

Grwinven can also output the ORL inventory format (in addition to I/O API format), which is also a SMOKE input format.

2.14.7. Using Smkmerge to apply the multiplicative and/or reactivity control matrices

The **Smkmerge** program is a second option for applying control matrices. **Smkmerge** can apply both multiplicative control matrices and reactivity control matrices; it is the only program that can do the latter. Because the overall purpose of **Smkmerge** is to create the model-ready input files for an AQM (as discussed in Section 2.18, "Creating model-ready emissions" [92], the result of using **Smkmerge** to apply control matrices is a controlled, model-ready file, as opposed to the controlled intermediate inventory file output by **Grwinven**.

Smkmerge is limited to applying one control matrix and one reactivity matrix for each SMOKE source category (area, mobile, or point). Therefore, if multiple reactivity control matrices need to be applied to create a single set of model-ready emission inputs, then the processing must be done as multiple SMOKE runs. This includes separating the sources that need different controls into separate inventory files and performing all SMOKE processing steps separately on the runs, using different reactivity controls for each. The resulting two or more sets of model-ready emissions should not have duplicate sources if the inventories were separated correctly at the start, and the model-ready files are combined using the **Mrggrid** program, as discussed in Section 2.18, "Creating model-ready emissions" [92].

2.15. Elevated-source processing

As introduced in Section 2.5.3, "Model-ready files" [21] and further explained in Section 2.8.5, "Point-source processing" [47], there are two major approaches to processing elevated point sources for air quality modeling. The first approach is to have SMOKE compute the layer assignments for the point sources; this method is used for the CMAQ and MAQSIP models. The second approach, which is used for the UAM models, REMSAD, and CAM$_X$, is to select specific sources as elevated and then create a special elevated-point-source file that contains the information needed so that the AQM can compute the plume rise. In both cases, users can select elevated sources specifically (in the second approach, that selection is mandatory). Also, PinG sources can be selected in both cases.

The two approaches have some steps that are the same and some that are different. The rest of this section is split into the following two subsections, one for each elevated-point-source processing approach:

1. Computing layer fractions for CMAQ and MAQSIP

2. Creating an elevated-source file for UAM, REMSAD, and CAM$_X$

Each subsection first overviews the steps in the approach, then gives more details on some of them.

2.15.1. Computing layer fractions for CMAQ and MAQSIP

The MAQSIP and CMAQ models require the layer fractions for elevated point sources to be computed by SMOKE. To do this, SMOKE performs the following steps:

- Uses the **Smkinven** program to import the annual, average-day, day-specific, and/or hour-specific emissions.

- Optionally uses the **Temporal** program to calculate hourly emissions if emissions will be used as a criterion for selecting elevated sources or PinG sources. For example, you may wish to select facilities with NO$_x$ emissions greater than 100 tons/day.

- Uses the **Elevpoint** program to select elevated and/or PinG sources. This step is required if modeling with PinG sources, but optional otherwise.

- Uses the **Elevpoint** program to create the STACK_GROUPS file, which is needed for processing PinG sources with CMAQ.

- Uses the **Laypoint** program to compute elevated plume rise for all elevated sources, and store the layer fractions for each source. This step can optionally read the output file from **Elevpoint** to identify the elevated sources, but otherwise will compute plume rise for *all* sources.

- Uses the **Smkmerge** program to combine the layer fractions with the hourly emissions to generate the model-ready output files and optionally generate the PinG hourly emissions file for CMAQ.

- Optionally uses the **Smkreport** program to report on elevated or PinG sources

2.15.1.1. Using Elevpoint

Elevpoint can select elevated and PinG sources using multiple criteria based on emissions values, emissions rank, stack parameters, plant numbers, and an analytical plume rise calculation. The elevated criteria and PinG criteria are provided to **Elevpoint** using a file called PELVCONFIG. You must configure this file to contain the criteria needed to select elevated and PinG source (if these selection are needed at all). If emissions values is one of the criteria, the **Elevpoint** program reads all of the hourly point-source files using the PTMPLIST file. This file is a list of all PTMP files that will be evaluated to determine which sources have maximum daily emissions that exceed the specified selection criteria or to determine the emissions rank. Only the maximum daily facility-total emissions can be used by **Elevpoint** to select sources based on emissions.

The elevated sources and PinG sources can each be selected using different criteria. In general, there are many more elevated sources than PinG sources for typical applications of SMOKE and AQMs. **Elevpoint** also permits you to group PinG sources and creates the STACK_GROUPS file so that the sources are treated as a single source in the PinG rise calculation by CMAQ. Grouping is useful to reduce the total number of PinG stacks processed by CMAQ (PinG processing is a computationally expensive calculation). Using grouping makes sense when several stacks at the same plant have the same, or nearly the same, stack parameters. When that is true, the emissions from the multiple stacks can be grouped and treated as a single

PinG stack. Finally, there are two SMOKE settings (`SMK_ELEV_METHOD` and `SMK_PING_METHOD`) that instruct **Elevpoint** and other SMOKE programs to actually use these criteria to select the elevated and PinG sources. If these settings are not set to "1" the elevated and PinG selections will not be made, and so will not affect any further processing steps.

2.15.1.2. Using Laypoint

In this CMAQ/MAQSIP approach, **Laypoint** uses gridded, hourly meteorological data and stack parameters to calculate the plume rise for all point-source emissions. The program's approach is based on the Briggs algorithm, as explained in detail in Section 6.3.1.1, "Plume rise calculation" [224], and provides the top and bottom heights of the plume. **Laypoint** uses these heights to compute the plumes' distributions into the vertical layers that the plumes intersect, using the pressure difference across each layer over the pressure difference across the entire plume as a weighting factor to make this calculation. This approach gives plume fractions by layer and source. Only these fractions are stored in the output file (`PLAY`) from the **Laypoint** program (not the emissions in each layer).

If explicit plume rise sources (e.g., wildfires with precomputed hourly plume rise) are included in the inventory, **Laypoint** will skip the plume rise calculation for these sources. Instead, it will use the hourly data from the SMOKE `PHOUR` intermediate file, which describe the fraction of emissions in layer 1 and the top and bottom of the plume. **Laypoint** will combine these data with the pressure weights used for all elevated point sources to compute the fraction of emissions to go into each layer.

2.15.1.3. Using Smkmerge

Smkmerge applies the layer fractions from **Laypoint** to the elevated sources to compute the emissions in each layer. This approach has the advantage of allowing you to avoid repeating the plume rise calculations for each control strategy or grid. If the `SMK_PING_METHOD` setting (discussed above) indicates that the special CMAQ PinG file should be created, the **Smkmerge** program will also output this special file, called the `PINGTS_L` file. It contains the hourly, speciated emissions for each PinG source (which could be a stack group, as explained above).

2.15.1.4. Optional use of Smkreport

If desired, **Smkreport** can apply the layer fractions and elevated or PinG statuses to the inventory to generate reports that include layer information and/or the elevated or PinG status. This reporting could be used, for example, to create a list of all PinG sources in the inventory, or to determine the elevated versus layer-1 emissions by state or SCC.

2.15.2. Creating an elevated-source file for UAM, REMSAD, and CAM$_X$

The other approach to modeling elevated sources is to create an elevated-point-source input file for one of the UAM models, REMSAD, or CAM$_X$. To do this, SMOKE performs the following steps:

- Uses the **Smkinven** program to import the annual, average-day, day-specific, and/or hour-specific emissions.

- Optionally uses the **Temporal** program to calculate hourly emissions if emissions will be used as a criterion for selecting elevated sources or PinG sources. For example, you may wish to select facilities with NO$_x$ emissions greater than 100 tons/day.

- Uses the **Elevpoint** program to select elevated and optionally PinG sources. Unlike processing for CMAQ or MAQSIP, this step is always required.

- Uses the **Elevpoint** program to create the `STACK_GROUPS` file, which is needed for creating the elevated-point-source file with **Smkmerge**.

- Uses the **Laypoint** program to compute elevated plume rise for explicit plume rise sources, and store the layer fractions for the explicit sources only.

- Uses the **Smkmerge** program to combine the `STACK_GROUPS` information, optional explicit plume rise information, and the hourly emissions to generate an ASCII elevated-point-source file with optional PinG flags.

2.15.2.1. Using Elevpoint

The primary difference between this approach and that for the CMAQ and MAQSIP models is that the **Elevpoint** processing step is required. Traditionally, elevated point sources have been selected for the UAM, REMSAD, or CAM$_X$ models using the analytical plume rise calculation. While **Elevpoint** can perform this calculation and use it exclusively to determine the elevated sources, you can also use the other selection criteria if desired: emissions values, emissions rank, stack parameters, and plant numbers. As described in Section 2.15.1, "Computing layer fractions for CMAQ and MAQSIP" [84], if the selection criteria include emissions, then the `PTMPLIST` file will be used to input all hourly emissions files for the entire modeling episode.

2.15.2.2. Using Laypoint

If explicit plume rise sources (e.g., wildfires with precomputed hourly plume rise) are included in the inventory, you must run the **Laypoint** program to compute the layer fractions for these sources only. This is the *only* reason **Laypoint** would be run in the UAM/REMSAD/CAM$_X$ processing approach, and it requires that the `EXPLICIT_PLUMES_YN` and the `HOUR_PLUMEDATA_YN` settings be set to Y. These settings cause **Laypoint** to write the layer fractions to the `PLAY_EX` file instead of to the usual `PLAY` file. For these explicit sources, **Laypoint** will skip the plume rise calculation. Instead, it will use the hourly data from the SMOKE `PHOUR` intermediate file containing the fraction of emissions in layer 1 and the top and bottom of the plume. **Laypoint** will combine these data with the pressure weights used for all elevated point sources to compute the fraction of emissions to go into each layer.

2.15.2.3. Using Smkmerge

The **Smkmerge** program looks for the `SMK_ASCIIELEV_YN` setting to determine whether the ASCII output file should be created. When this is set to Y, the output files from **Elevpoint** (`PELV` and `STACK_GROUPS`) are read to determine which sources should not be included in the 2-D emissions output file for point sources. The emissions from these sources are instead output to an ASCII elevated file along with stack parameters and locations so that the AQM can compute the plume rise.

If `EXPLICIT_PLUMES_YN` is also set to Y, **Smkmerge** will read the `PLAY_EX` file for the explicit plume sources. Since the UAM-based approach assumes that the AQM will compute the plume rise, SMOKE must manipulate the input file to trick the model into using precomputed plume rise. This is done by inserting fake stacks into the ASCII elevated file that extend to the center of each of the model layers and setting the stack parameters so that the plume rise calculation will keep the emissions associated with the fake stacks in the layer of the stack. As the emissions move from layer to layer, **Smkmerge** moves the reported emissions in the ASCII elevated file from fake stack to fake stack to represent the same behavior.

2.16. Mobile-source processing with MOVES

MOVES is the U.S. Environmental Protection Agency's (EPA) Motor Vehicle Emission Simulator. In the modeling process, the user specifies vehicle types, time periods, geographical areas, pollutants, vehicle

operating characteristics, and road types to be modeled. The model then performs a series of calculations, which have been carefully developed to accurately reflect vehicle operating processes (such as cold start or extended idle) and provide estimates of bulk emissions or emission rates.

An important feature of MOVES is that it allows users to choose between (1) the Inventory calculation type, which provides emission rates in terms of total quantity of emissions for a given time period; and (2) Emission Rate calculation type, which gives emission rates in terms of grams/mile or grams/vehicle/hour. For large-scale emissions modeling such as that needed for regional- and national-scale air quality modeling projects, it is desirable to use the Emission Rate calculation type, which populates emission rate lookup tables that can then be applied to many times and places, thus reducing the total number of MOVES runs required.

To reduce the time and effort and to help the user obtain more accurate modeling results, users need to prepare and post-process MOVES runs for a reference county (See Section 2.8.4.6, "Reference Counties" [41]) and reference fuel month (See Section 2.8.4.7, "Reference Fuel month" [42]) This approach consists of a set of scripts that automate the proper use of the Emission Rate calculations for the purpose of estimating mobile-source emissions for air quality (AQ) modeling.

Integrating MOVES into SMOKE modeling system consists of three major parts:

1) Meteorological data processing

- The meteorological data preprocessor program **Met4moves** prepares spatially and temporally averaged temperatures and relative humidity data to set up the meteorological input conditions for MOVES and SMOKE using the Meteorology-Chemistry Interface Processor (MCIP) output files.

2) MOVES model processing

- The MOVES Driver script : **"Runspec_generator.pl"** creates data importer files and the MOVES input file (runspec), which specifies the characteristics of the particular scenario to be modeled.

- The MOVES postprocessing scripts : **"Moves2smkEF.pl"** formats the MOVES emission rate lookup tables for SMOKE. **"gen_8digit_scc.pl"** generates an SCC mapping file used when importing activity data.

3) SMOKE model processing

- Spatially and temporally allocate mobile onroad activity data (i.e., Vehicle Mileage Traveled and Vehicle population).

- The MOVES postprocessing program, **Movesmrg**, estimates emissions from on-road mobile sources based on MOVES-based emission rate lookup tables and meteorology data from **Met4moves**.

- Creates hourly gridded speciated air quality model-ready input files.

- Produces various types of reports for users.

2.16.1. Meteorology Data Processing

With the specified reference county and reference fuel month approach for temperature and RH calculation for MOVES and SMOKE modeling systems, **Met4moves** uses hourly min/max temperatures and averaged RH over the spatial region that includes all of the inventory counties in a county group over the user-defined modeling period. **Met4moves** supports the averaging method (monthly or daily) to create min/max temperatures and averaged RH for all inventory counties in the county group(s). **Met4moves** determines the min/max grid cell temperatures and associated RH for both SMOKE and MOVES, and computes average

24-hour temperature profiles using Meteorology-Chemistry Interface Processor (MCIP) output files for use in MOVES. These **Met4moves** program is discussed in detail in Section 6.7, "**Met4moves**" [239].

The 24-hour temperature profiles are averaged over a user-specified time period and grid cells for all reference counties. For the MOVES Driver scripts, **Met4moves** outputs monthly average RH, min/max temperatures, and 24-hour temperature profiles in local time for all reference counties into one output file. For the SMOKE model, **Met4moves** outputs county-specific min/max temperatures and averaged RH values in local time for every inventory county and averaging period in the modeling inventory.

2.16.2. MOVES Model Processing

2.16.2.1. MOVES Driver Script

The inputs to the MOVES Driver Script (**Runspec_generator.pl**) include the temperature and humidity conditions output from **Met4moves** and two additional inputs, the RunControl and RepCounty files. The RunControl file contains pollutant selections and the file path location to the **Met4moves** output. The RepCounty file contains file path locations to user-created MOVES-formatted inputs for age distribution, fuel supply and formulation, inspection and maintenance programs, county level population and annual VMT for each reference county. Precise formats of the RunControl and RepCounty files can be found in the Section 8.9.2.2.2, "Reference County File for MOVES Driver Script" [429] and Section 8.9.2.2.1, "Run Control File for MOVES Driver Script" [428].

2.16.2.2. MOVES Post-processing Script

A MOVES Driver Script to automates the MOVES run setups and prepare two kinds of batch files. When launched, the batch files instruct MOVES to:

• import data into MySQL County Scale databases

• run MOVES for each runspec file

The resulting RatePerDistance (RPD), RatePerVehicle (RPV), RatePerProfile (RPP), and RatePerHour (RPH) tables contain all the conditions needed for regional modeling using SMOKE modeling system.

Once a MOVES batch run completes, MOVES populates the four output lookup tables with formats listed in Table 2.14, "MOVES Emission Rate Lookup Table (MOVES Format)" [89]. A MOVES Post-processing Script (**Moves2smkEF.pl**) in Perl interacts with MySQL to modify the default formats shown in Table 2.14, "MOVES Emission Rate Lookup Table (MOVES Format)" [89] into a text ASCII-format readable by SMOKE, shown in Table 2.15, "MOVES Emission Rate Lookup Table (SMOKE Format)" [90].

Table 2.14. MOVES Emission Rate Lookup Table (MOVES Format)

RatePerDistance (grams/mile)	RatePerVehicle (grams/vehicle/hour)	RatePerProfile (grams/vehicle/hour)	RatePerHour (grams/activity-hour)
MOVESScenarioID	MOVESScenarioID	MOVESScenarioID	MOVESScenarioID
MOVESRunID	MOVESRunID	MOVESRunID	MOVESRunID
yearID	yearID	temperatureProfileID	yearID
monthID	monthID	yearID	monthID
dayID	dayID	dayID	dayID
hourID	hourID	hourID	hourID
linkID	zoneID	pollutantID	linkID
pollutantID	pollutantID	processID	pollutantID
processID	processID	sourceTypeID	processID
sourceTypeID	sourceTypeID	SCC	sourceTypeID
SCC	SCC	fuelTypeID	SCC
fuelTypeID	fuelTypeID	modelYearID	fuelTypeID
modelYearID	modelYearID	Temperature	modelYearID
roadTypeID	Temperature	RatePerProfile	roadTypeID
avgSpeedBinID	RatePerVehicle		Temperature
Temperature			RatePerHour
relHumidity			
RatePerDistance			

The MOVES Post-processing Script converts the MOVES format Emission Rate Lookup Table to the SMOKE format Emission Rate Table by performing the following:

- Parses the state-county FIPS code from linkID, zoneID or TemperatureProfileID and stores it as a unique field, FIPS.

- Removes fields that are uninformative to SMOKE, including the MOVES source type, fuel type and road type, hourID in RatePerDistance, TemperatureProfileID in RatePerProfile.

- Reduces the output database table size by performing a cross-tab query on the pollutant emissions, listing each pollutant in a separate field rather than in a single column with a higher number of data records. The script also sorts the lookup tables by countyID, monthID and SCC for more efficient processing in SMOKE.

- Optionally calculates additional output pollutants or species by applying user-specified formulas to MOVES-created emission factors.

- Write out to the ASCII-formatted four processed SMOKE-ready MOVES lookup tables [RatePerDistance (RPD), RatePerVehicle (RPV), RatePerProfile (RPP), and RatePerHour (RPH)].

Table 2.15. MOVES Emission Rate Lookup Table (SMOKE Format)

RatePerDistance (grams/mile)	RatePerVehicle (grams/vehicle/hour)	RatePerProfile (grams/vehicle/hour)	RatePerHour (grams/activity-hour)
MOVESScenarioID	MOVESScenarioID	MOVESScenarioID	MOVESScenarioID
yearID	yearID	yearID	yearID
monthID	monthID	monthID	monthID
FIPS	dayID	dayID	FIPS
SCC	hourID	hourID	SCC
avgSpeedBin	FIPS	FIPS	temperature
temperature	SCC	SCC	CO
relHumidity	temperature (for each grid cell)	temperature (24hr temporal profile)	TOG
CO	CO	THC	BENZENE
TOG	NOX	TOG	NOX
BENZENE	PM10OC	VOC	VOC
...	...	...	...

2.16.3. SMOKE Model Processing

Once the **Met4moves** meteorology preprocessor and the MOVES model processing that results in the SMOKE-formatted emissions factor lookup tables are completed, we address the remaining major component of the SMOKE-MOVES tool: the SMOKE model processing step. The goals of this step are (1) to estimate emissions from on-road mobile sources based on MOVES-based emissions lookup tables and meteorology data, (2) to create hourly gridded speciated air quality model-ready input files, and (3) to produce various types of reports for the user.

As some readers are aware, MOBILE6 and MOVES are both vehicle emissions modeling systems used with SMOKE. However, they differ in their approaches to calculating off-network evaporative emissions. In MOBILE6, off-network emissions processes are calculated as emission factors in grams/mile, which is related to Vehicle Mileage Travelled (VMT). MOVES, on the other hand, uses the source (vehicle) type population (VPOP) to calculate start and off-network evaporative emissions, which are assigned to off-network emissions processes; these processes are hour-dependent due to VPOP (activity) assumptions built into the MOVES model. Thus, compared to the SMOKE-MOBILE6 approach, the SMOKE-MOVES approach requires additional vehicle population inventory data as input for estimating mobile-source emissions from off-network emissions processes. This requirement is reflected in the discussion below.

When processing mobile-source emissions from MOVES, SMOKE performs the following basic steps.

- **Smkinven** imports county-total VMT and average speed, and county-total vehicle hotelling hours (HOTELLING) by SCC for On-roadway Emission Processes and county-total vehicle population (VPOP) by vehicle type for Off-network Emission Processes.

- **Spcmat** computes the chemical speciation factors for each county, fuel type, source (=vehicle) type, road type, emission process, and pollutant, and stores the necessary factors for the VMT-to-species in RatePerDistance (RPD), HOTELLING-to-species in RatePerHour (RPH), and VPOP-to-species transformations in RatePerVehicle (RPV) and RatePerProfile (RPP).

- **Grdmat** allocates the county sources to grid cells and uses spatial surrogates to allocate county-total VMT, HOTELLING, and VPOP to grid cells, storing the factors needed for these allocations.

- **Temporal** computes hourly VMT and HOTELLING activity data for On-roadway Emission Processes (RPD and RPH tables). Off-network Emission Processes (RPV and RPP tables) do not require the **Temporal** program because vehicle population does not need to be temporally allocated.

The way the MOVES-generated emissions factor lookup tables are used varies according to whether SMOKE is modeling on-roadway emission processes or off-network emissions processes.

On-roadway emission processes: When estimates of all on-roadway emission processes except for extended idle exhaust are needed, SMOKE requires county-total VMT and average hourly speed (SPEED) inventory data as inputs to a SMOKE postprocessor called **Movesmrg** which is part of the SMOKE-MOVES tool. When extended idle exhaust process is estimated, counthy-total HOTELLING activity data is needed for **Movesmrg**. **Movesmrg** uses the SMOKE-ready MOVES RPD and RPH lookup tables as an input to estimate on-road sources emission. The key lookup fields for RPD are gridded hourly temperature and average hourly speed from the avgSpeedBinID field for RPD. SMOKE interpolates in the emission factors lookup table (in unit of grams/mile) based on gridded hourly temperature and average speed. Figure 2.13, "MOVES mobile RatePerDistance processing steps" [36] shows processing steps for on-roadway emissions processes in the SMOKE system using VMT and SPEED activity inventory data. For RPH, the gridded houlry temperature is used to interpolate in the emission factors lookup table (in unit of grams/hour).

Off-network emission processes: When estimates are needed for off-network emissions processes, including the off-network vapor venting emissions process, SMOKE uses county-total VPOP by vehicle type as input to **Movesmrg** together with the SMOKE-ready RatePerVehicle (RPV) and RatePerProfile (RPP) lookup tables. Figure 2.14, "MOVES mobile RatePerVehicle and RatPereProfile (off-network) processing steps" [36] shows processing steps for off-network emissions processes in the SMOKE modeling system using vehicle population activity inventory data. A significant difference in the processing steps between the on-roadway emissions processes (RPD table) and the off-network emissions processes (RPV and RPP tables) is that off-network emissions processing does not require the **Temporal** program step because vehicle population (VPOP) does not need to be temporally allocated. In the RPV table, gridded hourly temperature and hour of the day are the key lookup fields SMOKE uses to estimate hourly off-network emissions in unit of grams/vehicle/hour. For the evaporative fuel off-network vapor venting emissions process only, **Movesmrg** uses the RPP lookup table to estimate the emission rates based on the minimum and maximum temperatures computed by **Met4moves**.

For all SMOKE-ready three emission rate lookup tables (RPD, RPV, RPP), SMOKE performs linear interpolation when using them.

2.17. Biogenic processing

The SMOKE system uses a more advanced emissions modeling approach for biogenic processing than it uses for the other source types. For biogenic emissions, the temporal processing is a true simulation model driven by ambient meteorology and other data. SMOKE currently supports two methods for computing biogenic land use: BEIS2 and BEIS version 3.09 and version 3.14 (hereafter referred to as BEIS3). The overall processing performed by both these models is quite different from the processing done for anthropogenic source categories. BEIS2 and BEIS3 start with spatial allocation of land use data as the first processing step (which is the same as importing the raw inventory data for anthropogenic sources). They next compute normalized emissions for each grid cell and land use category. The final step is adjusting the normalized emissions based on gridded, hourly meteorology data and assigning the chemical species to output a model-ready biogenic emissions file. The following two subsections provide more detail about the two biogenic models in SMOKE and give references to additional detail elsewhere.

2.17.1. BEIS3 processing

SMOKE also includes two versions of BEIS3: version 3.09 and version 3.14. The concepts in BEIS3 are similar to those in BEIS2, except the program **Normbeis3** (instead of **Rawbio**) reads gridded land use data from three files in I/O API input format (BELD3_A, BELD3_B, and BELD3_TOT). Most users rely on the BELD3 database to generate these gridded land use files, but other land use databases may be used, provided the vegetation types and file formats are consistent with the BELD3_A, BELD3_B, and BELD3_TOT files. This database consists of high-resolution (1-km horizontal resolution) land use data for 230 land use types. **Normbeis3** creates normalized biogenic emissions for both winter and summer. The **Tmpbeis3** program reads the MCIP meteorology data and makes these adjustments; these include the effects of temperature and solar radiation. The winter emission factors are to be used during the "winter" period, which is defined as being after the first date of freezing ground temperature and before the last date of freezing ground temperature. The SMOKE **Metscan** program can determine this time period and create a gridded file that indicates whether each grid cell is in a winter or summer period for each day of the year. Using this file, which is optional, will cause **Tmpbeis3** to read and use both the winter and summer normalized emissions files from **Normbeis3**. Whether the winter or summer normalized emissions are used for a given grid cell and hour is set by the output from the **Metscan** program. Users can define the chemical species that are output from **Tmpbeis3** using the speciation profile file, GSPRO. In this file, you can set an environment variable (BIOG_SPRO) that indicates which speciation profile code should be used for biogenic emissions. Note that this GSPRO file is different than the one used for BEIS2. More information on **Normbeis3** and **Tmpbeis3** is provided in Section 6.12, "**Normbeis3**" [256] and Section 6.17, "**Tmpbeis3**" [299].

2.18. Creating model-ready emissions

Creating emissions with SMOKE that are ready for input to an AQM must always include merging the hourly emissions created during temporal processing with the gridding matrices and the speciation matrices. In addition, for point sources for CMAQ or MAQSIP, creating the model-ready emissions must also include merging with the layer fractions (see Section 2.15.1, "Computing layer fractions for CMAQ and MAQSIP" [84]), and for UAM-based models it must include creating the ASCII elevated file (see Section 2.15.2, "Creating an elevated-source file for UAM, REMSAD, and CAM$_X$" [85]). The **Smkmerge** program performs these processing steps using vector-matrix multiplication to combine the matrices and layer fractions with the hourly emissions vectors from the **Temporal** program.

If the overall SMOKE processing setup includes running the nonroad mobile category (or other source categories) as separate runs (see Chapter 4, *Using SMOKE Scripts* [117], then **Smkmerge** cannot be used to combine all source categories into a single output file. Instead, the **Mrggrid** program would combine the model-ready files from the individual source categories; for example, from separate SMOKE (including **Smkmerge**) runs for stationary area/nonpoint, nonroad mobile, windblown dust, wildfire, on-road mobile, and point sources. There is no limit to the number of model-ready files that **Mrggrid** can combine into a single model-ready file, and the input files can be 2-D or 3-D.

Smkmerge can be run for any or all SMOKE source categories, but it can use only one of each SMOKE inventory type (area, biogenic, mobile, and point source) per run. You can run it to create model-ready files for only one SMOKE source category (area, biogenic, mobile, or point), or you can run it to create both the individual and combined model-ready files. The following list indicates the modes in which **Smkmerge** can be run:

- Run for SMOKE area sources to create gridded, hourly, speciated emissions in moles/hour or moles/second. Can be used for all area sources and/or nonroad mobile sources.

- Run for SMOKE mobile sources to create gridded, hourly, speciated emissions in moles/hour or moles/ second.

- Run for SMOKE point sources to create 3-D gridded, hourly, speciated emissions in moles/hour for MAQSIP and moles/ second for CMAQ, *or* run to create 2-D gridded, hourly, speciated emissions in moles/hour for UAM, REMSAD, or CAM$_X$ and an ASCII elevated-point-source file.

- Run to convert the units and calculate state and county totals of biogenic emissions output by the **Tmpbio** or **Tmpbeis3** programs.

- Run to perform any combination of the previously listed steps simultaneously and create a combined model-ready file that includes multiple source categories. In this mode, only one each of SMOKE area, mobile, point, and biogenic sources can be included. The same output units must be used for all source categories in a single run.

When creating model-ready emissions for any of the anthropogenic source categories, you may choose to apply one or more control matrices to the emissions to create controlled model-ready emissions. For each source category (area, mobile, or point), you can apply one multiplicative control matrix and one reactivity control matrix per run per source category. **Smkmerge** is the only way you can apply the reactivity control matrix to the inventory, while the multiplicative control matrix can be applied by either **Smkmerge** or the **Grwinven** program.

Many processing steps in SMOKE are independent of one another; for example, chemical speciation and temporal allocation can change without affecting one another. This independence means that when one step changes, another step does not need to be rerun in many cases. However, because **Smkmerge** combines the data from all of these processing steps to create the model-ready emissions, if one of the earlier steps changes, then the merging step must also be rerun. This includes rerunning **Smkmerge** to generate model-ready files, and if the **Mrggrid** program was used, also rerunning that to merge data from multiple source categories together.

Smkmerge also has the ability to input hourly emissions by day of the week and reuse days that are the same. For example, it can input separate hourly emissions files for Monday, a weekday, Saturday, and Sunday, and use these four files to generate model-ready emissions for every day in an entire month. This is accomplished using the MRG_BYDAY SMOKE option, described further in Section 6.14, "**Smk-merge**" [271]. Special treatment can also be given to holidays in this case, since users generally wish to model holidays differently than other days.

2.19. Creating model-ready emissions using MOVES lookup tables

Creating emissions with SMOKE using MOVES lookup tables (i.e., RatePerDistance [RPD], RatePerHour [RPH], RatePerVehicle [RPV] and RatePerProfile [RPP]) must always include merging the gridding matrices from **Grdmat** and the chemical speciation matrices from **Spcmat**. In addition, for RPD emissions based on VMT data by source for CMAQ or MAQSIP, creating the model-ready emissions must also include merging with the hourly VMT from **Temporal**. The **Movesmrg** program performs these processing steps using vector-matrix multiplication to combine the matrices and the hourly emissions vectors to create CMAQ-ready gridded and speciated hourly emissions input data.

If the overall SMOKE processing setup includes running the nonroad mobile category (or other source categories) as separate runs (see Chapter 4, *Using SMOKE Scripts* [117], then **Movesmrg** cannot be used to combine all source categories into a single output file. Instead, the **Mrggrid** program would combine the model-ready files from the individual source categories; for example, from separate SMOKE (including **Movesmrg**) runs for RPD, RPV and RPP mobile sources. There is no limit to the number of model-ready files that **Mrggrid** can combine into a single model-ready file, and the input files can be 2-D or 3-D.

Movesmrg can be used only for MOVES-based mobile SMOKE source categories. It can use only one of each MOVES lookup table (RPD, RPH, RPV and RPV) per run. You can run it to create model-ready files for only one SMOKE source category (RPD, RPH, RPV, and RPP). The following list indicates the modes in which **Movesmrg** can be run:

- Run for MOVES RPD mobile sources based on VMT data by vehicle and road types to create gridded, hourly, speciated emissions in moles/hour or moles/second.

- Run for MOVES RPH mobile sources based on vehicle hotelling (HOTELLING) hours activity data by vehicle and road types to create gridded, hourly, speciated emissions in moles/hour or moles/second.

- Run for MOVES RPV mobile sources based on Vehicle population (VPOP) data by vehicle to create gridded, hourly, speciated emissions in moles/hour or moles/second.

- Run for MOVES RPP mobile sources based on Vehicle population (VPOP) data by vehicle to create gridded, hourly, speciated emissions in moles/hour or moles/second.

- Run to create daily total emissions report by county, by state, and by SCC in the unit of tons/day or tons/hour.

Many processing steps in SMOKE are independent of one another; for example, chemical speciation and temporal allocation can change without affecting one another. This independence means that when one step changes, another step does not need to be rerun in many cases. However, because **Movesmrg** combines the data from all of these processing steps to create the model-ready emissions, if one of the earlier steps changes, then the merging step must also be rerun. This includes rerunning **Movesmrg** to generate model-ready files, and if the **Mrggrid** program was used, also rerunning that to merge data from multiple source categories together.

Movesmrg also has the ability to input hourly emissions by day of the week and reuse days that are the same. For example, it can input separate hourly emissions files for Monday, a weekday, Saturday, and Sunday, and use these four files to generate model-ready emissions for every day in an entire month. This is accomplished using the MRG_BYDAY SMOKE option, described further in Section 6.8, "**Movesmrg**" [242]. Special treatment can also be given to holidays in this case, since users generally wish to model holidays differently than other days.

2.20. Quality assurance

Quality assuring SMOKE emissions includes a combination of (1) steps performed by SMOKE programs and (2) postprocessing steps performed by the user.

The SMOKE components that play a role in quality assurance consist of the following:

1. The various SMOKE programs perform file format checks of all input files to ensure that the files can be read. The programs write errors and warnings if files cannot be read properly.

2. SMOKE gives error and warning messages about inventory data that are not complete or are invalid, and about problems or possible problems combining the inventory data with the support files.

3. Core SMOKE programs create reports, such as the area-to-point report provided by **Smkinven** and the control reports provided by **Cntlmat**.

4. The **Smkreport** program reports emissions totals at various levels of data aggregation. This reporting capability currently allows you to generate reports of emissions by source, SCC, region (e.g., state, county, or user-defined region), road class, layer, hour, grid cell, speciation profile, gridding surrogate code, temporal profile, and elevated status. The most powerful reporting feature is that you can combine

these reporting resolutions in *any* combination. In addition, reports can be created at each stage of processing (import, gridding, speciation, temporal allocation, layer assignment) or any combination of stages.

Smkreport can combine information from any SMOKE intermediate files (e.g., intermediate inventory file, speciation matrix, gridding matrix) to create emissions reports. One input file to **Smkreport**, called the REPCONFIG file, instructs **Smkreport** on how many and which reports to create. The REPCONFIG file contains a series of instructions that can be set by the user to control the contents of the reports. More details about **Smkreport** and the REPCONFIG are provided in Section 7.2, "**Smkreport** Program" [305] and Section 7.3, "REPCONFIG Input File" [308].

The second major component of quality assuring the SMOKE emissions processing involves users taking steps to evaluate the information/reports provided by SMOKE. These steps include:

1. Check that the correct settings have been selected in the run scripts, including the settings that control which SMOKE programs are run.

2. Check the log files from all SMOKE programs for errors and warnings. Errors will keep the programs from running successfully, so the source of the error must be identified and repaired. Warnings may indicate that a problem exists that needs to be addressed, or warnings can be ignored if they are not something that will impact the results for the particular inventory of interest.

3. Compare the emissions totals provided by **Smkreport** (e.g., by state and county) to totals of the emissions inventories computed outside of SMOKE. Also, compare the emissions totals from SMOKE between each of the processing stages. For example, for area sources, compare the emissions after inventory import, gridding, chemical speciation, temporal allocation, and final merge to ensure that the emissions are consistent from step to step. This involves some subjectivity because the emissions do in fact change from step to step, and the magnitude of those changes depends on the support input files SMOKE uses with the inventory.

4. Check that the correct chemical speciation profiles, temporal profiles, and gridding surrogates were applied, using reports that provide this information from **Smkreport**.

5. Perform other specific checks of **Smkreport** outputs, such as ensuring that the correct major point sources are in the inventory, comparing population-normalized emissions among the counties, and checking stack parameters from point sources.

6. Ensure that the emissions data look reasonable by viewing them in the Visualization Environment for Rich Data Interpretation (VERDI).

Chapter 7, *SMOKE Quality Assurance* [305] provides much additional detail about how to proceed with quality assuring your inventories and emissions processing.

Chapter 3. SMOKE Directory Structure

Table of Contents

3.1. Introduction

The primary user interface for SMOKE is a series of UNIX scripts. Because the basic SMOKE installation and instructions use these scripts, a key factor in using SMOKE is understanding the directory structure used for reading input files and writing output files. Additionally, it is critical that users know where to find the run scripts and other files that must be configured for a specific application of SMOKE. Section 3.2, "Directory structure" [97] of this chapter describes the default directory structure.

Because SMOKE is based on UNIX scripts, it can also be reconfigured to use a different directory structure. This is useful if SMOKE is to be used from the MIMS interface, incorporated into another modeling system, or modified to suit the needs of a particular user. In Section 3.3, "How to change the default directory structure" [115], we describe what must be done to reconfigure the default directory structure to meet such needs.

In this chapter, we describe the default directory structures set automatically by the Assigns file provided with the default installation of SMOKE. We discuss a basic directory structure and an advanced structure that follow the same approach and can easily be adjusted using the Assigns file. For example, you can change a scenario name in the Assigns file, and this will automatically change all of your intermediate and output directory names. However, these two directory structures are only examples; as noted above, SMOKE is flexible enough to use any structure one wants to create. You could rework the Assigns file (or even completely remove it) if so desired, although this would be a much more complex process than changing a few environment variables. We have developed the SMOKE default directory structure based upon experience gained from many years of emissions modeling; we hope that SMOKE users will find it useful.

In addition to a default directory structure, SMOKE comes with a test case including example data files. Section 4.2, "Test case descriptions" [117] provides descriptions about these files including where the files were obtained. These descriptions are intended to provide you with some background on the data files and a starting point for developing your own modeling files.

3.2. Directory structure

The SMOKE directory structure is the organization of UNIX directories and subdirectories (essentially the same as "folders" on Windows computers). These directories are defined and created using the SMOKE Assigns file. By "defined", we mean that environment variables are set to the actual path name for each of the directories, and their relationships to one another are implicit in these definitions. As will be described in more detail in Chapter 4, *Using SMOKE Scripts* [117], you should create an Assigns file for each modeling case to organize your modeling runs. If created properly, the different Assigns files will cause the intermediate and output directories to be different for each case; this prevents overwriting files from one case with files from another. Some parts of the directory structure are shared by all modeling cases because they are usually the same (e.g., SMOKE support files), but the intermediate and output files are not assumed to be the same because they often differ between cases.

3.2.1. Example Assigns file

Before we can proceed with describing the scripts, we need to introduce the Assigns file. Detailed documentation on this file is provided in Section 4.2.2.2, "Assigns files" [120]. For the purposes of the discussion here, we provide an example Assigns file to help clarify what it is. Because this file is quite lengthy, we have only included excerpts. In this subsection, the Assigns file is simply a set of **setenv** UNIX commands and calls to helper scripts.

The arrangement within the default Assigns file provided with SMOKE is:

- Shell command (e.g., **#!/bin/csh -fx** for C shell)

- Descriptive global comments

- Settings for the naming roots

- Settings for the episode dates and times

- Settings for the directory structure and use of the set_dirs.scr helper script

- Settings for SMOKE input and output files and calls to other helper scripts

The first settings section is used for naming roots. These settings allow users to set the inventory input and output identifiers, the base-case scenario names, and the grid and chemical mechanism names (see Chapter 4, *Using SMOKE Scripts* [117] for more details). The following lines provide the naming roots used in the default Assigns file provided with SMOKE, ASSIGNS.nctox.cmaq.cb4pm25_wtox.us12-nc. This file and the helper scripts can be found in the $ASSIGNS directory, which we will define in Section 3.2.2, "The basic SMOKE directories" [99].

```
#
## I/O Naming roots
#
    setenv INVID     nctox         # Inventory input identifier
    setenv INVOP     nctox         # Base year inventory output name
    setenv INVEN     nctox         # Base year inventory name with version

    setenv ABASE     nctox         # Area base case output name
    setenv BBASE     nctox         # Biogenics base case output name
    setenv MBASE     nctox         # Mobile base case output name
    setenv PBASE     nctox         # Point base case output name

    setenv EBASE     nctox         # Output merged base case name

    setenv METSCEN       12km              # Met scenario name
    setenv GRID          us12-nc           # Gridding root for naming
    setenv IOAPI_GRIDNAME_1 US12NC_66X52 # Grid selected from GRIDDESC file
    setenv IOAPI_ISPH    19                # Specifies spheroid type associated with gri
    setenv SPC           cmaq.cb4p25_wtox # Speciation type
```

The next section of the file includes the settings for the episode:

```
## Mobile episode variables
    setenv EPI_STDATE 2001191     # Julian start date
    setenv EPI_STTIME  000000     # start time (HHMMSS)
    setenv EPI_RUNLEN 0480000     # run length (HHHMMSS)
```

```
    setenv EPI_NDAY   2            # number of full run days

## Per-period environment variables
    setenv G_STDATE   2001191      # Julian start date
    setenv G_STTIME   000000       # start time (HHMMSS)
    setenv G_TSTEP     10000       # time step   (HHMMSS)
    setenv G_RUNLEN   250000       # run length (HHMMSS)
    setenv ESDATE   20010710       # Start date of emis time-based files/dirs
    setenv MSDATE   20010710       # Start date of met time-based files
    setenv NDAYS           1       # Duration in days of each emissions file
    setenv MDAYS           1       # Duration in days of met time-based files
    setenv YEAR         2001       # Base year for year-specific files
```

As described in Chapter 4, *Using SMOKE Scripts* [117], these settings control both the way SMOKE runs (e.g., SMOKE uses EPI_STDATE, the start date of the episode, to decide what day to start the simulation) and the way SMOKE names files (e.g., ESDATE is used to help name time-dependent files).

The next section of the Assigns file is used to set the user-defined input and output directory paths. It uses a helper script called set_dirs.scr that applies the initial settings made by the Assigns file to set environment variables for the directories, as described in Section 3.2.2, "The basic SMOKE directories" [99]. The set_dirs.scr script also verifies that all of the directories are there and that the permissions of the directories are group readable and writable.

```
## User-defined I/O directory settings
    setenv SMK_SUBSYS  $SMK_HOME/subsys              # SMOKE subsystem dir
    setenv SMKROOT     $SMK_SUBSYS/smoke             # System root dir
    setenv SMKDAT      $SMK_HOME/data                # Data root dir
    setenv ASSIGNS     $SMKROOT/assigns              # SMOKE assigns files

<omitted section>

## Set dependent directory names
#
source $ASSIGNS/set_dirs.scr
```

Now that we have provided a little background on Assigns files, we will spend the rest of this chapter describing the SMOKE directory structure. We will return to discussion of the Assigns file and scripts in Chapter 4, *Using SMOKE Scripts* [117]. It is important to understand the directory structure information in the rest of this chapter before moving on to Chapter 4, *Using SMOKE Scripts* [117].

3.2.2. The basic SMOKE directories

By default, SMOKE is installed in a directory called SMK_HOME (the complete process for installing SMOKE is described in Chapter 12, *Downloading, Installing, and Compiling SMOKE* [495]). As shown in the previous section, SMK_HOME is an environment variable on which other SMOKE directories depend; it is the root directory of all other SMOKE directories. When you install SMOKE, you must define and create your own SMK_HOME directory as part of the installation process. Since SMK_HOME is an environment variable, you can choose whatever name you like; we cannot prescribe a name for you because each UNIX user has a different computer and disk names. One recommendation that we can make is that SMK_HOME be set close to the bottom of a directory structure, such as a short directory name like /models. Using a short directory name will limit the overall directory lengths for some of the subdirectories deep within the SMOKE directory structure. The SMK_HOME directory used for SMOKE can be the same directory used to install VERDI [http://www.verdi-tool.org] or the I/O API [http://www.baronams.com/products/ioapi/].

Figure 3.1, "Basic SMOKE directories: the first three levels of the directory structure" [100] shows the first three layers of the SMOKE directory structure. All of the directories in the figure are created during the SMOKE installation process. From the SMK_HOME directory stems the data directory, the scripts directory, and the subsys directory. The data directory corresponds to the environment variable SMKDAT, while the subsys directory corresponds to SMK_SUBSYS; the environment variables are shown in parentheses underneath the actual directory names in the figure. The scripts directory does not have an associated environment variable since we did not feel that one was necessary. The subsys directory is discussed in more detail within this subsection. The data directory will be described more fully in Section 3.2.3, "Locations of input files" [102], Section 3.2.4, "Locations of intermediate files for base and growth/control cases" [106], Section 3.2.5, "Locations of model-ready output files for base and growth/control cases" [112], and Section 3.2.6, "Locations of reports" [114], when we describe the locations of input, intermediate, output, and report files. We will not discuss the scripts directory since it only contains two general setup scripts that should not need to be modified under normal SMOKE use.

Figure 3.1. Basic SMOKE directories: the first three levels of the directory structure
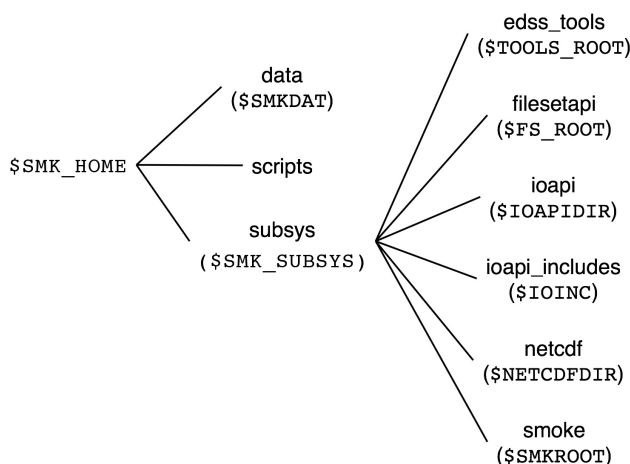
The environment variables shown in parentheses are available to you once you invoke the Assigns file from the command line. The run scripts that come with SMOKE invoke the Assigns file to set up the environment variables automatically. To manually invoke the Assigns file, you must go to the $SMKROOT/assigns directory and type

source <Assigns file name>

where <Assigns file name> is replaced with your specific Assigns file name. Note that SMOKE only works with variants of the C-shell including **csh** and **tcsh**. If you are running a different shell (such as **bash**), you will need to switch into a C-shell before sourcing the Assigns file or running any SMOKE scripts.

Once this command has been used in a UNIX window (e.g., **xterm**, **terminal**, **console**), then the environment variables are available at the command line or to a script that uses the Assigns file. What this means is that the UNIX environment "remembers" the setting of the environment variable. To access the contents of the variable - in this case, a path name - the dollar sign ($) must be used in front of the variable from a UNIX prompt or script. From the prompt, this is most useful for changing directories. For example, after the Assigns file has been invoked, the command

cd $SMKDAT

will take you to the $SMK_HOME/data directory, no matter where else in the directory structure you were located when the command was issued. Similarly, the Assigns file can use these environment variables for setting the paths for the input and output files.

As shown in Figure 3.1, "Basic SMOKE directories: the first three levels of the directory structure" [100], the subsys directory (creating during installation of SMOKE) contains six subdirectories, each with the following purposes:

- edss_tools: Contains the source code, makefiles, and library object files for the EDSS Tools library, necessary for compiling SMOKE programs.

- filesetapi: Contains the source code, makefiles, and library object files for the FileSetAPI library, necessary for compiling SMOKE programs.

- ioapi: A placeholder location for the I/O API library object files. When compiling SMOKE, the makefiles look here for the I/O API library and the SMOKE scripts look here for the I/O API utility programs. The files can be linked to the main I/O API installation which can be located elsewhere. Please see Chapter 12, *Downloading, Installing, and Compiling SMOKE* [495] for more information.

- ioapi_includes: Location for the I/O API include files, needed for compiling the FileSetAPI library, EDSS Tools library, and SMOKE libraries and programs. All makefiles that come with SMOKE look here for the I/O API include files.

- netcdf: A placeholder location for the NetCDF library object files. When compiling SMOKE, the makefiles look here for the NetCDF library, but the actual library can be linked to the main NetCDF installation, which can be elsewhere on your computer. Please see Chapter 12, *Downloading, Installing, and Compiling SMOKE* [495] for more information.

- smoke: This is the main SMOKE directory. It contains the Assigns files (one per modeling case), SMOKE source code, makefiles, scripts, and executables.

Figure 3.2, "Subdirectories of the main SMOKE directory" [101] shows the subdirectories inside the main SMOKE directory, smoke; these are all created during the installation of SMOKE. These subdirectories contain the Assigns files, source code and makefiles, scripts, and executables.

Figure 3.2. Subdirectories of the main SMOKE directory

```
                                              assigns
                                             ($ASSIGNS)

                                              doc

                    smoke                     $EDSS_EXE
                  ($SMKROOT)                  ($SMK_BIN)

                                              scripts
                                             ($SCRIPTS)

                                              src
```
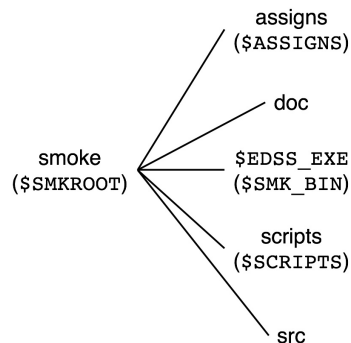
These five subdirectories have the following purposes:

- assigns: Contains the Assigns files and all helper scripts for the Assigns files.

- doc: Contains the files that the SMOKE installation processes use to check that the installation and testing of SMOKE have worked properly.

- `$SMOKE_EXE`: This directory contains the executables for your particular system. Different directory names apply to different platforms (e.g., `Linux2_x86_64pg`, `IRIXn32f90`, `SunOS5f90`). For more information about the various platforms for which SMOKE is available, see Chapter 12, *Downloading, Installing, and Compiling SMOKE* [495].

- `scripts`: Contains run scripts and other scripts

  - `scripts/install`: Scripts to install SMOKE and check the installation

  - `scripts/make`: Scripts for compiling SMOKE

  - `scripts/run`: Default run scripts and associated helper scripts

- `src`: Contains subdirectories that store the source code and makefiles for each SMOKE program:

  - `src/biog`: **Normbeis3**, **Tmpbeis3**

  - `src/cntlmat`: **Cntlmat**

  - `src/emmod`: Fortran 90 modules used by multiple SMOKE programs

  - `src/emqa`: **Smkreport**

  - `src/emutil`: **Aggwndw**, **Beld3to2**, **BlueSky2Inv**, **CEMScan**, **Geofac**, **Invsplit**, **Layalloc**, **Metcombine**, **Metscan**, **Pktreduc**, **Smk2emis**, **Surgtool**, **Uam2ncf**

  - `src/grdmat`: **Grdmat**

  - `src/inc`: Include files shared by multiple SMOKE programs

  - `src/lib`: SMOKE library

  - `src/movesmrg`: **Movesmrg**, **Met4moves**

  - `src/point`: **Elevpoint**, **Laypoint**

  - `src/smkinven`: **Grwinven**, **Smkinven**

  - `src/smkmerge`: **Mrgelev**, **Mrggrid**, **Smkmerge**

  - `src/spcmat`: **Spcmat**

  - `src/temporal`: **Temporal**

3.2.3. Locations of input files

Input directories, which contain SMOKE input files, are one of the subdirectory types that make up the `$SMKDAT` directory shown in Figure 3.1, "Basic SMOKE directories: the first three levels of the directory structure" [100] as `$SMK_HOME/data`. In this section, we provide the structure of these input directories.

Figure 3.3, "SMOKE input files directory structure" [103] shows all of the input directories that stem from `$SMKDAT`. In the diagram, note how some paths depend on environment variables set in the Assigns file. For example, the `INVID` environment variable listed in Section 3.2.1, "Example Assigns file" [98] is defined by the user (as we will describe in Chapter 4, *Using SMOKE Scripts* [117]) to label the inventory name. This variable is used to build the name of the `INVDIR` directory as follows:

`INVDIR = $SMKDAT/inventory/$INVID`

So, if a user decided to label the inventory "nei99", then the Assigns file would have `INVID` = nei99 and the `INVDIR` directory would be `$SMKDAT/inventory/nei99`. Another example of this in the figure is the `SMK_M6PATH` directory. The full path depends upon the setting of the `MSCEN` variable; if `MSCEN` is set to "case1", then `SMK_M6PATH` would be `$SMKDAT/run_case1/static/m6`.

In some cases, we do not have environment variables associated with a directory name. One example is the `$SMKDAT/inventory` directory. In these cases, we did not assign variables simply because they were not very useful and we have to limit the number of SMOKE environment variables. Most operating systems have a limit on the number of environment variables (usually 256), so when the variables were not really useful we did not assign them. If desired, you can add environment variables to the Assigns file and/or the `set_dirs.scr` helper script.

Figure 3.3. SMOKE input files directory structure
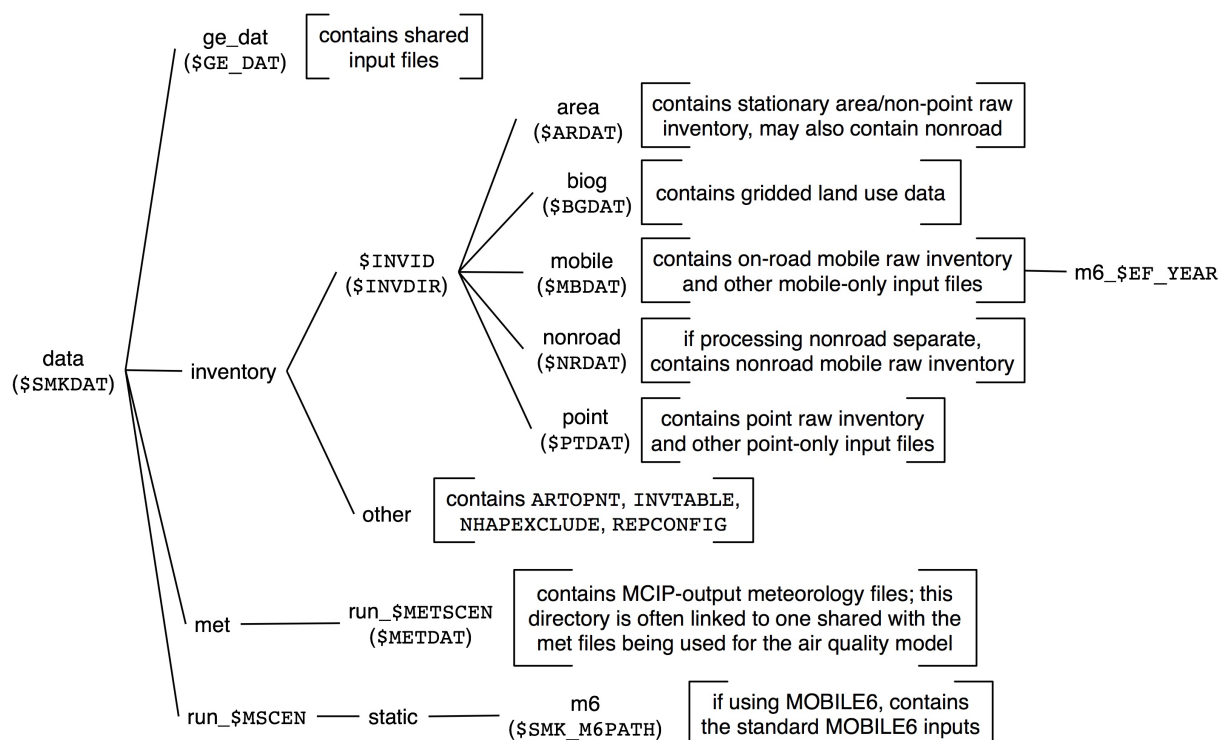
In the list below, we describe the purpose of each of these directories, along with the file names and in some cases file descriptions for those files that you may need to change. You can refer to Chapter 8, *SMOKE Input Files* [345] for more information about each of these files (look under the logical file names). Chapter 4, *Using SMOKE Scripts* [117] provides step-by-step instructions on how to create and install each of the files listed.

- `ge_dat`: This directory contains the cross-reference files, profile files, spatial surrogates, and other files that are likely to be shared among multiple cases of processing inventories with SMOKE. The logical file names for files included in this directory are listed here; many physical files could be available in the `ge_dat` directory for each file type. For example, you may have dozens of surrogate files for different modeling grids and several speciation profiles for different model-mechanism combinations. More detailed descriptions of these files are in Chapter 8, *SMOKE Input Files* [345]; descriptions of the example files provide with SMOKE are in Section 4.2, "Test case descriptions" [117].

- `BGPRO`: The spatial surrogates file biogenics (for computing county-total emissions with **Smk-merge**).

- `SRGPRO_PATH`: The location of the spatial surrogates file for nonpoint (stationary area)/nonroad and on-road mobile sources.

- `AGREF`, `MGREF`: The spatial cross-reference files for nonpoint (stationary area)/nonroad and on-road mobile sources. These two logical file names can all reference the same physical file.

- `ATPRO`, `MTPRO`, `PTPRO`: The temporal profiles files for nonpoint (stationary area)/nonroad, on-road mobile, and point sources. These three logical file names can all reference the same physical file.

- `ATREF`, `MTREF`, `PTREF`: The temporal cross-reference files for nonpoint (stationary area)/nonroad, on-road mobile, and point sources. These three logical file names can all reference the same physical file.

- `B3FAC`: The BEIS3.14 normalized emission factors.

- `B360FAC`: The BEIS3.60 normalized emission factors.

- `BIOSEASON`: I/O API file indicating which grid cells in a domain are winter and which are summer for biogenics modeling purposes, for each day of the year or episode.

- `COSTCY`: The country, state, and county information file.

- `GRIDDESC`: The grid definitions file.

- `SRGDESC`: The surrogate definitions/description file.

- `GSREF`, `GSPRO`, `GSCNV`: The chemical speciation cross-reference file, speciation profiles file, and pollutant-to-pollutant conversion file (e.g., VOC-to-TOG factors), respectively.

- `HOLIDAYS`: The dates that SMOKE should process as holidays and the day of the week to use for each date instead of its actual day of the week (the substitute is usually a Saturday or Sunday).

- `PROCDATES`: The sequential/non-sequential user-specified dates that SMOKE should process.

- `PSTK`: The point-source replacement stack parameters.

- `SCCDESC`, `SRGDESC`, `ORISDESC`, `SICDESC`, `MACTDESC`, `NAICSDESC`: The descriptions of each known SCC, surrogate code, ORIS ID, SIC code, MACT code, and NAICS code.

- `VNAMMAP`: The name-mapping file for transforming the I/O API file model-species names to the names used by the model (e.g., renaming CD for REMSAD to CD-1).

- `inventory`: The location of all inventory files for each inventory case (as defined by the `$INVID` environment variable in an Assigns file), land use data, and other data.

  - `$INVID`: This directory name depends on the name given to the inventory in the Assigns file. (This is the case for any directory name in the figure that is preceded with "$" to indicate that it is an environment variable.) The files in the subdirectories are specific to your inventory files and modeling case. This directory contains subdirectories for each of the major source categories by default:

    - `area`: Contains the nonpoint or stationary area inventories. Optionally, it may contain nonroad mobile inventories. Can include *any* inventory that will be processed as a SMOKE area source (e.g. criteria, particulates, and/or toxics inventories). Files should include:

- `ARINV` (raw area-source inventory file). If this is a list file, the files listed within it should also be included in this directory.

- `ARDAY` (raw day-specific area-source inventory file). If this optional file is a list file, the files listed within it should also be included in this directory.

- `ARHOUR` (raw hour-specific area-source inventory file). If this optional file is a list file, the files listed within it should also be included in this directory.

- `GCNTL` (control packet file). The growth and control packet files for each growth or control scenario. The default naming scheme for these files is described in the default SMOKE scripts.

- `biog`: Contains the gridded BELD3 and/or BELD4 land use data. These files are listed in the Assigns files as `BELD3_A`, `BELD3_B`, and `BELD3_TOT` for BEIS3.14. This directory could optionally contain the gridded BELD4 land use data if proceessing using BEIS3.60); this file is listed in the Assigns file as `BELD4`.

- `mobile (MOVES)`: Contains the on-road mobile inventory and other on-road mobile-specific input files. Files should include:

  - `GCNTL` (control packet file). The growth and control packet files for each growth or control scenario. The default naming scheme for these files is described in the default SMOKE scripts.

  - `MBINV` (raw on-road mobile-source inventory file). This is a FF10 activity inventory format for VMT, SPEED and VPOP inventory data. EAA - put in a link to input file description.. If this is a list file, the files listed within it should also be included in this directory.

  - `MCXREF` (mobile-source county cross-reference). For MOVES modeling only, contains the reference county assignments (see Section 2.8.4.6, "Reference Counties" [41] for an explanation of reference counties).

  - `MEPROC` (mobile-source processing file). For MOVES modeling only, contains the pollutants to be modeled by MOVES by the emission processes (e.g., exhaust, evaporative, break and tire wear).

  - `SPDPRO` (speed profiles file). For MOVES modeling only, contains the optional weekday/weekend hour-specific speed profiles for use of MOVES on-roadway emission rates.

- `point`: Contains the point-source inventory and other point-specific input files. Files should include:

  - `CEMSUM` (CEM summary file). When processing CEM data, this file contains a summary of a year's worth of CEM data and is produced by **CEMScan**.

  - `GCNTL` (control packet file). The growth and control packet files for each growth or control scenario. The default naming scheme for these files is described in the default SMOKE scripts.

  - `PELVCONFIG` (elevated-source selection configuration file). When using the **Elevpoint** program to select elevated and PinG sources, this file is required to specify the selection criteria.

  - `PTDAY` (raw day-specific point-source inventory file). If this optional file is a list file, which is typically the case, the files listed within it should also be included in this directory.

- PTHOUR (raw hour-specific point-source inventory file). If this optional file is a list file, which is typically the case, the files listed within it should also be included in this directory.

- PTINV (raw point-source inventory file). If this is a list file, the files listed within it should also be included in this directory.

- other: Contains files that are associated with the various inventories, but which may be shared by more than one source category. These files are:

  - ARTOPNT (area-to-point file). Contains the area-to-point assignments by SCC with the point locations for each county in which a SMOKE area source will be assigned to point locations.

  - INVTABLE (inventory table file). Contains a list of the valid inventory pollutants and whether or not emissions associated with each pollutant should be kept when reading the inventory files, among other purposes.

  - NHAPEXCLUDE (non-HAP exclusions file). Contains the SCCs that will be excluded from a NONHAPVOC or NONHAPTOG calculation when integrating toxics and criteria inventories.

  - REPCONFIG (report configuration file). Contains instructions used by **Smkreport** to configure reports. The SMOKE installation includes only the REPCONFIG files for the default reports included with the SMOKE default scripts.

- met/run_$METSCEN: This directory contains the meteorology files that are needed for running (1) biogenic emissions; (2) on-roadway and off-network mobile sources with MOVES using gridded, hourly meteorology data; or (3) point sources for the CMAQ or MAQSIP models. Meteorology data are required in all of these cases. SMOKE can use output files from CMAQ's MCIP and MCIP2 meteorology processors or from the Meteorology Coupler (MCPL) processor. SMOKE automatically checks for the different variable names that may be used for the same meteorology data in the different file formats. See Section 8.8, "Meteorology Files" [403] for more information about the meteorology data needed by SMOKE.

3.2.4. Locations of intermediate files for base and growth/control cases

The intermediate files for SMOKE are located in three primary directories for each case: the inventory output directory ($INVOPD), the scenario directory ($SCENARIO), and the static files directory ($STATIC). All intermediate file locations are subdirectories of the $SMKDAT directory shown in Figure 3.1, "Basic SMOKE directories: the first three levels of the directory structure" [100] as $SMK_HOME/data. Note that Figure 3.4, "Locations of intermediate and output files for SMOKE base case, basic configuration" [107], Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110], and Figure 3.6, "Locations of intermediate and output files for SMOKE future-/past-year case" [111] show both intermediate and output directories. Here we focus only on the intermediate directories; the output directories are discussed in Section 3.2.5, "Locations of model-ready output files for base and growth/control cases" [112].

3.2.4.1. Basic configuration for base case

Figure 3.4, "Locations of intermediate and output files for SMOKE base case, basic configuration" [107] shows the SMOKE intermediate directories for a basic configuration of a base case. In this configuration, the Assigns file variables ASCEN, BSCEN, MSCEN, PSCEN, and ESCEN are all set to the same value. These variables are the scenario names for the four processing categories (area, biogenic, mobile, and point) and the combined category (used when combining emissions from multiple source categories).

Having these variables set to the same value allows a simpler structure in which the `run_$ESCEN` directory is the same as the `run_$ASCEN`, `run_$BSCEN`, `run_$MSCEN`, and `run_$PSCEN` directories. In the figure, this single directory is shown simply as `run_$ESCEN`.

The `$BSCEN` setting for biogenic sources is often set to a different scenario name, usually simply "beld2" or "beld3". This is because there are not very many scenarios to choose from for biogenic emissions, since users almost always want to use one of the standard approaches and there are usually no control strategies considered for biogenic emissions. In this case, the `run_$BSCEN` directory is a separate branch in the directory structure that stores just the biogenic files for use by both base case and future-year control scenarios. For the nctox test case, we have not used this approach since the biogenic data is specific to the nctox modeling domain.

Figure 3.4. Locations of intermediate and output files for SMOKE base case, basic configuration
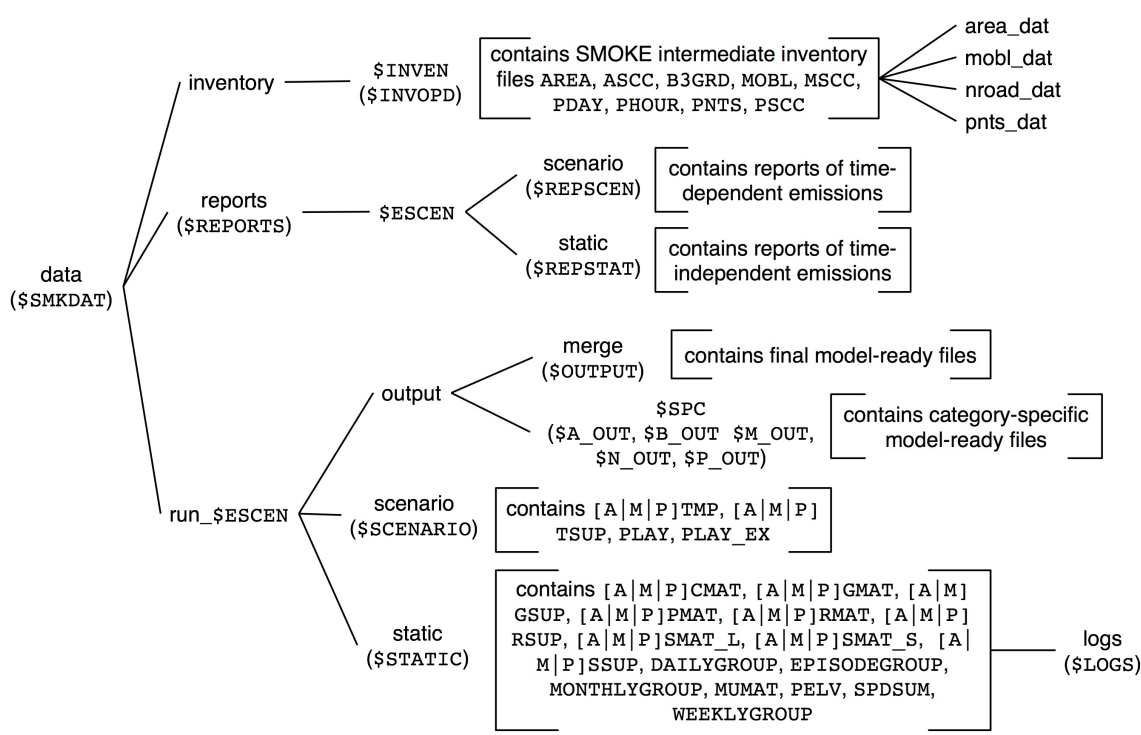
The following list describes the intermediate files that are associated with each of the intermediate directories `$INVOPD`, `$SCENARIO`, and `$STATIC`. You can refer to Chapter 9, *SMOKE Intermediate Files* [461] for more information about each of the intermediate files (look under the logical file names).

- `inventory/$INVEN`: This directory contains the SMOKE intermediate inventory files, including the directories associated with them. These files are the outputs from **Smkinven**, and **Tmpbeis3**. The outputs from **Smkinven** are map-formatted (see below for more information) while the biogenic outputs are I/O API NetCDF format. The subdirectories `area_dat`, `mobl_dat`, `nroad_dat`, and `pnts_dat` contain the files that store the emissions values for each pollutant in the inventory. The intermediate files in this directory include:

    - `AREA`: Map-formatted SMOKE intermediate files for nonpoint/stationary area sources (from the `ARINV` input file) and/or nonroad mobile sources (from either the `ARINV` input file or from the `NRINV` input file if nonroad mobile sources are being run separately from other SMOKE area sources). Example file names include `area.map.$INVOP.txt` and `nroad.map.$INVOP.txt`.

The data from the inventories is stored in several different files, both ASCII and I/O API format. The map-formatted intermediate files are used to organize and keep track of these separate files. See Chapter 9, *SMOKE Intermediate Files* [461] for an example and further discussion of the intermediate inventory files. For the `AREA` file, the additional inventory files `area.ncf` and `asrc.txt` are stored in the `$INVOPD` directory. The pollutant-specific files are stored in the `area_dat` directory. For the `NROAD` file, the files are named `nroad.ncf` and `nsrc.txt` and the pollutant-specific files are placed in the `nroad_dat` directory.

- `ARDAY`: The day-specific SMOKE area-source intermediate inventory.

- `ARHOUR`:The hour-specific SMOKE area-source intermediate inventory.

- `ASCC`: List of area-source SCCs in the inventory. Can include `ASCC.$INVOP.txt` and `NSCC.$INVOP.txt` (e.g., `ASCC.nei99.txt` and `NSCC.nei99.txt`).

- `B3GRD`: Normalized gridded biogenic emissions.

- `MOBL`: Map-formatted SMOKE intermediate file for on-road mobile sources. Example file name is `mobl.map.$INVOP.txt`. The additional inventory files are named `mobl.ncf` and `msrc.txt`, and the pollutant-specific files are placed in the `mobl_dat` directory.

- `MSCC`: List of on-road mobile-source SCCs in the inventory; e.g., `MSCC.$INVOP.txt`.

- `PDAY`: The day-specific SMOKE intermediate inventory.

- `PHOUR`: The hour-specific SMOKE intermediate inventory.

- `PNTS`: Map-formatted SMOKE intermediate file for point sources. Example file name is `pnts.map.$INVOP.txt`. The additional inventory files are named `pnts.ncf` and `psrc.txt`, and the pollutant-specific files are placed in the `pnts_dat` directory.

- `PSCC`: List of point-source SCCs in the inventory; e.g., `PSCC.$INVOP.txt`.

- `reports`: Discussed in Section 3.2.6, "Locations of reports" [114]

- `run_$BSCEN/output/$SPC`: Contains the hourly, gridded, speciated biogenic emissions files. These are listed in this subsection for intermediate files because in some cases these are not ready for the AQM, because the units need to be converted. **Tmpbeis3** can output emissions in either moles/hr or moles/s. The CMAQ model needs data in units of moles/s.

- `run_$ESCEN/output`: Discussed in Section 3.2.5, "Locations of model-ready output files for base and growth/control cases" [112]

- `run_$ESCEN/scenario`: Contains the intermediate hourly emissions files and other time-dependent files. The intermediate files in this directory include:

  - `ATMP, MTMP, PTMP`: The area/nonroad, mobile, and point hourly emissions created by the **Temporal** program.

  - `ATSUP, MTSUP, PTSUP`: The area/nonroad, mobile, and point temporal allocation supplementary files.

  - `PLAY`: The point-source hourly layer fraction for each source.

  - `PLAY_EX`: The point-source hourly layer fractions for *explicit* plume rise sources when processing for UAM, CAM$_X$, or REMSAD.

- `run_$ESCEN/static`: Contains the matrices and other time-independent files. The intermediate files in this directory include:

  - `ACMAT, MCMAT, PCMAT`: The area/nonroad, mobile, and point multiplicative control matrices.

  - `AGMAT, MGMAT, PGMAT`: The area/nonroad, mobile, and point gridding matrices.

  - `AGSUP, MGSUP`: The area/nonroad and mobile supplementary gridding files.

  - `APMAT, MPMAT, PPMAT`: The area/nonroad, mobile, and point growth matrices.

  - `ARMAT, MRMAT, PRMAT`: The area/nonroad, mobile, and point reactivity control matrices.

  - `ARSUP, MRSUP, MRSUP`: The area/nonroad, mobile, and point reactivity supplementary files.

  - `ASMAT_L, MSMAT_L, PSMAT_L`: The area/nonroad, mobile, and point mole-based speciation matrices.

  - `ASMAT_S, MSMAT_S, PSMAT_S`: The area/nonroad, mobile, and point mass-based speciation matrices.

  - `ASSUP, MSSUP, PSSUP`: The area/nonroad, mobile, and point supplementary speciation files.

  - `PELV`: The point-source elevated- and PinG-source selection file

  - `$LOGS`: Contains the log files created when the SMOKE programs are run

3.2.4.2. Advanced configuration for base case

Instead of using the basic configuration technique of setting the scenario name variables (`ASCEN`, `BSCEN`, `MSCEN`, `PSCEN`, and `ESCEN`) all to the same value (Section 3.2.4.1, "Basic configuration for base case" [106]), you can set up an Assigns file to use different values for one or more of these variable. In this more advanced configuration, separate directories are created and used for intermediate and output files for each source category (examples of when this would be useful are given in the bullets below). Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110] illustrates the directory structure resulting from this approach; however, for simplicity's sake, we have not included all of the files in this structure. The files are the same as in Figure 3.4, "Locations of intermediate and output files for SMOKE base case, basic configuration" [107], with the exception that files specific to a source category go to their category-specific directories, based on the values given to `ASCEN`, `BSCEN`, `MSCEN`, and/or `PSCEN`. Note that the nonpoint (stationary area) and nonroad mobile sources would need to use the same scenario name, even if they are processed separately, because the nonroad mobile `static` and `scenario` directories are set using the same `ASCEN` variable as for other area sources in the default Assigns files. Having separate source-category-specific directories does not change the number of intermediate or output files, it just rearranges the files and results into fewer intermediate, output, and/or report files in each directory.

This approach is particularly useful in the following cases:

- **Using toxic inventories for some source categories but not others:** In this case, different scenario names (`ASCEN`, `MSCEN`, `PSCEN`, and `ESCEN`) can be used to distinguish each case and the final-merged files (though it is not required to do this). For example, if you were running toxics for on-road mobile sources only, `MSCEN` could be set to "case1_tox", while `ASCEN` and `PSCEN` could be set to just "case1" and `ESCEN` could be "case1_all". If you would prefer instead to name all of these "case1" and simply remember that on-road mobile sources include toxics, then that would be fine as well.

- **Using the Mrggrid program to combine model-ready files from several previously run scenarios:**
  For example, you might want to combine the future-year base case for all source categories except nonroad with a future-year control case for nonroad, as part of an evaluation of the impact of the nonroad control case. If three previous Assigns files were set up using the basic structure for three different control cases (case1, case2, and case3), and a new case is needed that uses area sources from case1, on-road mobile sources from case2, and point sources from case3, you could set up a new Assigns file that sets ASCEN to case1, MSCEN to case2, and PSCEN to case3. When you invoke that new Assigns file, all of the environment variables needed by **Mrggrid** for combining the model-ready files from the three different cases would be set: AGTS_L would be based on case1, MGTS_L would be based on case2, and PGTS3D_L would be based on case3. See Section 3.2.5, "Locations of model-ready output files for base and growth/control cases" [112] for more information on the model-ready files.

Figure 3.5. Locations of intermediate and output files for SMOKE base case, advanced configuration
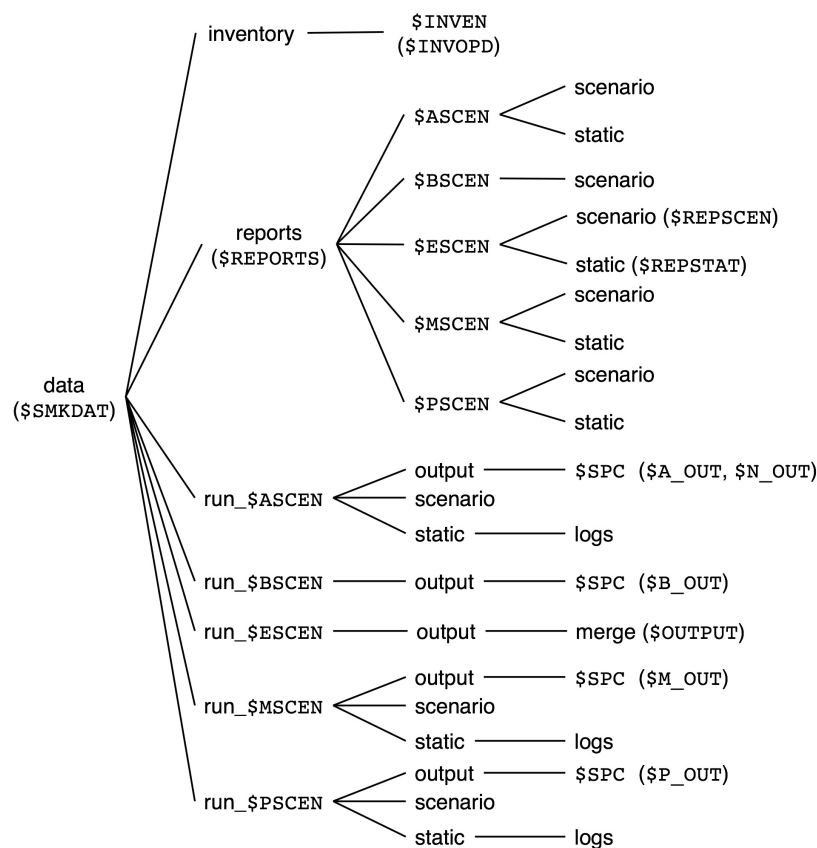
3.2.4.3. Configuration for growth and/or control case

The directory structure changes for growth and control cases. The default growth and control scripts have additional settings that impact the intermediate directories when running SMOKE. These settings are described in more detail in Chapter 4, *Using SMOKE Scripts* [117], but we introduce them here because they are closely linked to the directory structures used by SMOKE. The default configuration of SMOKE uses different directories for files impacted by growth and control so that they will be separated from the base-year files; this helps keep users from accidentally confusing their base and grown and/or controlled files.

The key script environment variables that impact the directory structures are:

- CNTLCASE: Sets the name of the control case that will impact the directory names. Also is used by the scripts to set the required name of the GCNTL files containing the packets for defining growth and controls for the **Cntlmat** program.

- FYEAR: Sets the future year with a four-digit year. The last two digits of the year are used to set the internal variable fyr2. For example, if FYEAR is set to 2018, fyr2 will be 18. FYEAR is also used by the scripts to set the required name of the GCNTL files containing the packets for defining growth and controls for the **Cntlmat** program.

- SMK_FUTURE_YN: When set to Y, this setting causes the Assigns file to change the SMOKE intermediate inventory file names by inserting $fyr2 into the individual file names. The Assigns file also changes the intermediate and output paths by appending $fyr2 to $ASCEN, $MSCEN, $PSCEN, and $ESCEN (affecting the $STATIC and $SCENARIO directories and file names of intermediate files). This is illustrated in Figure 3.6, "Locations of intermediate and output files for SMOKE future-/past-year case" [111].

- SMK_CONTROL_YN: When set to Y, this setting causes the Assigns file to change the SMOKE intermediate inventory file names by inserting $CNTLCASE into the individual file names. The Assigns file also changes the intermediate and output paths by appending $CNTLCASE to $ASCEN, $MSCEN, $PSCEN, and $ESCEN (affecting the $STATIC and $SCENARIO directories and file names of intermediate files).

Figure 3.6. Locations of intermediate and output files for SMOKE future-/past-year case
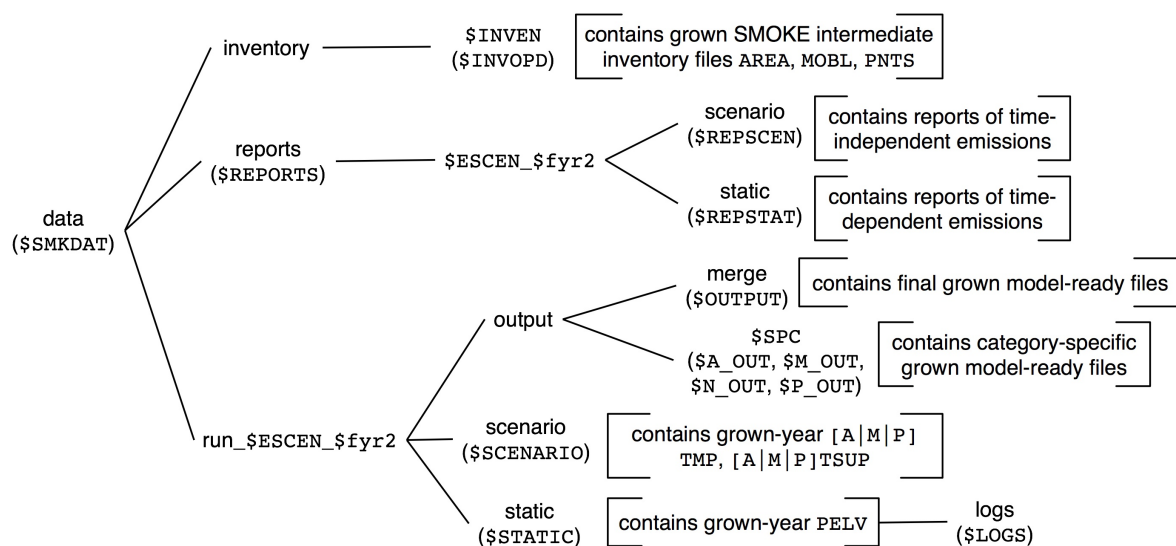
The $SCENARIO directory for the future case is now a subdirectory of data/run_$ESCEN_$fyr2. The ATMP, MTMP, and PTMP files are stored in this directory, because these depend on the future-year emissions inventory. Although the temporal profile assignments stored in the ATSUP, MTSUP, and PTSUP files may not have changed from the base case, these files are written out for the future-/past-year case to be consistent with the output files created by **Temporal** for the base case.

The $STATIC directory for the future case is now a subdirectory of data/run_$ESCEN_$fyr2. However, unlike Figure 3.4, "Locations of intermediate and output files for SMOKE base case, basic configuration" [107], Figure 3.6, "Locations of intermediate and output files for SMOKE future-/past-year case" [111] indicates that only the PELV file is stored in this directory. This is because SMOKE permits reusing the base-year gridding matrices and speciation matrices for growth and control cases. The growth and control matrices will also be stored in the base-year $STATIC directory. Therefore, the only file that

is stored in the future-year $STATIC directory is the PELV file, which can depend on the emissions values in the future year. For example, if the elevated-source selection criteria were based on the future-year daily maximum emissions, then this PELV file would be different than the base-year file. It is therefore most safely stored in a directory that is named with the future-year (or control case) information.

When using controls only (no growth), $SCENARIO and $STATIC will be subdirectories of data/run_$ESCEN_$CNTLCASE. For the case of growth and controls, the directory will be named data/run_$ESCEN_$fyr2_$CNTLCASE.

3.2.5. Locations of model-ready output files for base and growth/control cases

3.2.5.1. Configuration for base case

Figure 3.4, "Locations of intermediate and output files for SMOKE base case, basic configuration" [107], Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110], and Figure 3.6, "Locations of intermediate and output files for SMOKE future-/past-year case" [111] show the locations of the output files as well as the intermediate files. The output files go to the subdirectory of the data/run_$ESCEN directory called output. This output directory has two subdirectories: the first is set by the $SPC environment variable from the Assigns file that sets the model and chemical mechanism being used for a SMOKE run; the second is named merge. There can be just one set of these subdirectories (as in the basic configuration shown in Figure 3.4, "Locations of intermediate and output files for SMOKE base case, basic configuration" [107]) or multiple sets (as in the advanced configuration shown in Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110]).

The $SPC subdirectory contains the category-specific model ready files, such as the model-ready files for nonpoint (stationary area) only, nonroad mobile only, etc. The merge subdirectory contains the final model-ready files. The following list provides the logical names and descriptions of each of the files in the basic configuration case.

- run_$ESCEN/output/merge:

  - **For CMAQ or MAQSIP:**

    EGTS3D_L: CMAQ or MAQSIP 3-D hourly, gridded, speciated emissions file, excluding PinG emissions (from the **Smkmerge** program).

  - **For CMAQ only:**

    PINGTS_L: Optional file that includes the hourly emissions for PinG stacks (or grouped stacks) for each hour (from **Smkmerge**).

    STACK_GROUPS: Optional file that identifies PinG stacks (or grouped stacks), their locations, and stack parameters (from the **Elevpoint** program).

  - **For UAM, REMSAD, or CAM$_X$:**

    EGTS_L: I/O API version of the 2-D merged low-level, hourly, gridded, speciated emissions file (from **Smkmerge**).

    EGTS_UAM: Fortran binary 2-D merged low-level hourly, gridded, speciated emissions file (from the **Smk2emis** program).

ELEVTS_L: ASCII elevated-point-source stack parameters and hourly, speciated emissions (from **Smkmerge**).

- `run_$ESCEN/output/$SPC`:

  - `AGTS_L` and `AGTS_S`: Nonpoint (stationary area) and optionally nonroad mobile-source hourly, gridded, speciated emissions (from the **Smkmerge** program).

  - `B3GTS_L` and `B3GTS_S`: Biogenic hourly, gridded, speciated emissions in moles/hour and tons/hour (from the **Tmpbeis3** program).

  - `B3GTS_L_O` and `B3GTS_S_O`: **Tmpbeis3** output with the correct units needed by the AQM (from the **Smkmerge** program).

  - `MGTS_L` and `MGTS_S`: On-road mobile-source hourly, gridded, speciated emissions (from **Smkmerge**).

  - `NGTS_L` and `NGTS_S`: Nonroad mobile-source hourly, gridded, speciated emissions in moles/hour and grams/hour (these files are written by **Smkmerge** as an `AGTS_L` or `AGTS_S` file, with the scripts changing the files to the `NGTS` name).

  - `PGTS_L` and `PGTS_S`: Point-surce hourly, gridded, speciated emissions (from **Smkmerge**). This file would be created when modeling for UAM, REMSAD, or CAM_X, which need the low-level point sources but not the 3-D point sources. This file would not include the elevated-point-source emissions if it were created using the `SMK_ASCIIELEV_YN` = Y setting.

  - `PGTS3D_L` and `PGTS3D_S`: Point-source 3-D, hourly, gridded, speciated emissions (from **Smkmerge**). This file would be created for CMAQ and MAQSIP only, using the elevated layer fraction, which requires the option `MRG_LAYERS_YN` = Y.

3.2.5.2. Configuration for growth and/or control case

The output file locations for the growth and control cases follow the same paradigm described in Section 3.2.4.3, "Configuration for growth and/or control case" [110] for the intermediate files. The `$fyr2` and/or `$CNTLCASE` environment variable values are inserted into the directory structure, as shown in Figure 3.6, "Locations of intermediate and output files for SMOKE future-/past-year case" [111]. The output directories are then reset as follows:

Future/past-year case (no controls):

- `run_$ESCEN_$fyr2/output/merge`: Final model-ready output files. The files are the same as those in the base case, but with future-year emissions; see Section 3.2.5.1, "Configuration for base case" [112] for more information.

- `run_$ESCEN_$fyr2/output/$SPC`: Source-category hourly, gridded, speciated files. In the advanced configuration, the `$ASCEN`, `$MSCEN`, or `$PSCEN` values could be different from the `$ESCEN` value, resulting in separate directories (as shown in Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110], but with the `$fyr2` field included in the path name. The files are the same as those in the base case, but with future-year emissions; see Section 3.2.5.1, "Configuration for base case" [112] for more information.

Control case (no growth):

- `run_$ESCEN_$CNTLCASE/output/merge`: Final model-ready output files. The files are the same as those in the base case, but with controlled emissions; see Section 3.2.5.1, "Configuration for base case" [112] for more information.

- `run_$ESCEN_$CNTLCASE/output/$SPC`: Source-category hourly, gridded, speciated files. In the advanced configuration, the `$ASCEN`, `$MSCEN`, or `$PSCEN` values could be different from the `$ESCEN` value, resulting in separate directories (as shown in Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110], but with the `$CNTLCASE` field included in the path name. The files are the same as those in the base case, but with controlled emissions; see Section 3.2.5.1, "Configuration for base case" [112] for more information.

Future/past-year and control case:

- `run_$ESCEN_$fyr2_$CNTLCASE/output/merge`: Final model-ready output files. The files are the same as those in the base case, but with future-year, controlled emissions; see Section 3.2.5.1, "Configuration for base case" [112] for more information.

- `run_$ESCEN_$fyr2_$CNTLCASE/output/$SPC`: Source-category hourly, gridded, speciated files. In the advanced configuration, the `$ASCEN`, `$MSCEN`, or `$PSCEN` values could be different from the `$ESCEN` value, resulting in separate directories (as shown in Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110], but with the `$fyr2` and `$CNTLCASE` fields included in the path name. The files are the same as those in the base case, but with future-year, controlled emissions; see Section 3.2.5.1, "Configuration for base case" [112] for more information.

3.2.6. Locations of reports

The report output directory names follow the same paradigm as the intermediate and model-ready output directory names. "Reports" as defined for this directory structure are the report files output from the **Cntlmat**, **Smkinven**, **Smkmerge**, and **Smkreport** programs. More information on these programs can be found in Section 6.2, "**Cntlmat**" [213], Section 6.13, "**Smkinven**" [260], Section 6.14, "**Smkmerge**" [271], and Section 7.2, "**Smkreport** Program" [305].

As shown in Figure 3.4, "Locations of intermediate and output files for SMOKE base case, basic configuration" [107], the two main report directories for the basic configuration are the `data/reports/$ESCEN/scenario` and `data/reports/$ESCEN/static` directories. The former directory is for time-dependent reports, and the latter is for time-independent reports. Because the `BSCEN` variable is often different from the `ESCEN` variable, a different report directory is available for the biogenic emissions: `data/reports/$BSCEN/scenario`, which contains the only biogenic reports available (created by the **Smkmerge** program).

Figure 3.5, "Locations of intermediate and output files for SMOKE base case, advanced configuration" [110] shows the report directories for the advanced configuration. The source-category-specific variables `ASCEN`, `BSCEN`, `MSCEN`, and `PSCEN` appear here, in addition to the `ESCEN` variable. In the default setup of SMOKE, **Smkmerge** uses separate source-category-specific directories for the reports, as defined in the `$ASSIGNS/setmerge_files.scr` script. For **Smkreport**, the `$SCRIPTS/run/qa_run.csh` script sends the default reports to the `$REPSTAT` and `$REPSCEN` directories (which use the `ESCEN` variable in their path names). Both of these scripts are straightforward to change to accommodate user preferences.

Figure 3.6, "Locations of intermediate and output files for SMOKE future-/past-year case" [111] shows the report directories for the basic configuration of future/past-year and/or control case reports. The `$REPSTAT` and `$REPSCEN` directories use the year and/or control case name (from `$fyr2` and `$CNTLCASE`) to reset the report directory names, in the same way as for the intermediate and model-ready output files.

The **Smkmerge** program has numerous logical file names that can be used as output files (too many to list here). The reports include state and/or county totals for each day run by **Smkmerge**. The form of the output logical file names is `REP[A|B|M|P|E]G[T][S][C][_S|_L]`. The A, B, M, P, or E in the file represents

area, biogenic, mobile, point, or "every" emissions, respectively. **Smkmerge** can optionally report temporalized emissions (T), speciated emissions (S), and/or controlled emissions (C). The _S or _L suffix represents mass-based or mole-based reports, respectively (e.g., tons/day or moles/day).

The **Smkreport** program uses logical file names in the REPCONFIG file to determine report names. The /NEWFILE/ packet controls the logical file names, and the SMOKE script qa_run.csh (for default reporting) or the main run scripts control the values of these logical file names and hence the file paths. There are too many types of reports that **Smkreport** can create for us to list them all here with their appropriate files. In general, the time-independent reports go to the $REPSTAT directory and the time-dependent reports go to the $REPSCEN directory.

3.3. How to change the default directory structure

The default Assigns files and scripts contain settings that allow users to modify the directories that will be used when running SMOKE. The basic structure has been developed to suit the needs of a variety of applications, from short, single-grid, base case modeling needs to annual, multigrid needs for many control strategies. The details of working within the default structure by changing Assigns file and script settings are provided in Chapter 4, *Using SMOKE Scripts* [117]. The default structures that we have described represent our recommended way of running SMOKE. Although this structure has been quite useful for many people, it could be a burden to others. In this section, we describe ways in which users can change the default structure to meet their specific needs.

The SMOKE programs themselves do not require any specific directory structure because all input and output files depend only on environment variables, which can be set in any pattern. We have created a default directory structure so that other users can benefit from our experience in emissions modeling and using SMOKE, and to provide an initial structure that is useful. With a few exceptions for intermediate files, the SMOKE programs also do not require specific file names, only specific environment variables; but again, we have configured SMOKE to have a default structure for these names.

In principle, you can change the location of every single input file that SMOKE uses. The Assigns files and scripts have been implemented using one structure, but changing that structure is only a matter of finding some other method of providing the file names to the programs. This can include changing the Assigns files and scripts, or removing the Assigns files altogether and creating a script that sets all of the input and output file names and directories and calls the programs. These modifications can be as extensive or minimal as needed to handle a specific limitation or problem with the default structure.

If slight modifications to the structures are all that is needed, then the best approach is to modify a copy of the default the Assigns file. For example, if you wish to have the nonroad mobile sources' intermediate and output files written to a different directory than the stationary area sources' files, you would add an NSCEN environment variable to parallel the area-source variable ASCEN, and change the input, intermediate, and output file locations and names to use this setting.

Chapter 4. Using SMOKE Scripts

Table of Contents

4.1. Introduction

In this chapter, we provide information about using the UNIX scripts that come with SMOKE to run the test cases and adapting the scripts to run any case. There are several features of these scripts that specifically support SMOKE v1.5 and later, and using scripts developed for earlier versions of SMOKE will not work without adaptations. We provide a section in this document that explains what changes need to be made to users' existing scripts to allow them to work with SMOKE. Alternatively, current users can adapt the new example scripts to their case. New users are advised to create their scripts based on the example scripts, following the instructions included in Section 4.4, "How to use SMOKE" [125] this chapter.

For all SMOKE users, a critical file to use and know about is the Assigns file. If you have followed the SMOKE installation instructions, then your Assigns file will be located in the directory: `$SMK_HOME/subsys/smoke/assigns` where the `SMK_HOME` directory is set in your `.cshrc` file in your home directory, during the SMOKE installation process.

When using SMOKE, the first thing that you should do is go to the Assigns directory and invoke the Assigns file for the case that you will be working with. This step needs to be taken in every UNIX window from which you intend to run SMOKE.

To invoke the Assigns file, use the following commands:

`cd $SMK_HOME/subsys/smoke/assigns` *(this changes to the correct directory)*

`source ASSIGNS.nctox.cmaq.cb05_soa.us12-nc` *(this invokes the default assigns file)*

Doing this will set a large number of environment variables that you will then have available for navigating the SMOKE directory structure. Table 4.1, "Variable in Assigns file for naming files and directories" [128] and Table 4.2, "Variables in Assigns file for controlling time period and grid of processing" [130] in Section 4.4.1, "Change Assigns file to set scenario names, grid names, and other case-specific configuration information" [126] list those environment variables. The environment variables used to build some of the directory names are also set by the user in the Assigns file and described in that section.

4.2. Test case descriptions

Many of the support files used in the test cases are based on the EPA's NEI modeling platform, available online at EPA's Emission Modeling Clearinghouse (http://www.epa.gov/ttn/chief/emch/ [http://www.epa.gov/ttn/chief/emch/index.html]).

4.2.1. nctox case for area, nonroad, mobile, biogenic and point sources

The nctox case provides users with an example of using both criteria and toxics emissions inventories. The purposes of this case include:

- Providing a test case for SMOKE on users computers. It is not recommended for actual user's emissions modeling work.

- Provide example scripts that users may adapt to their own cases

This case has the following major features. It runs for July 10th and 11th, 2005 on a 12-km Lambert Conformal grid over North Carolina. The spatial extent of this case has been limited to provide a relatively fast test case, and the domain is a subdomain of EPA's 12-km national grid. The case includes toxics processing for stationary area, nonroad mobile, and point sources.

Figure 4.1. nctox domain
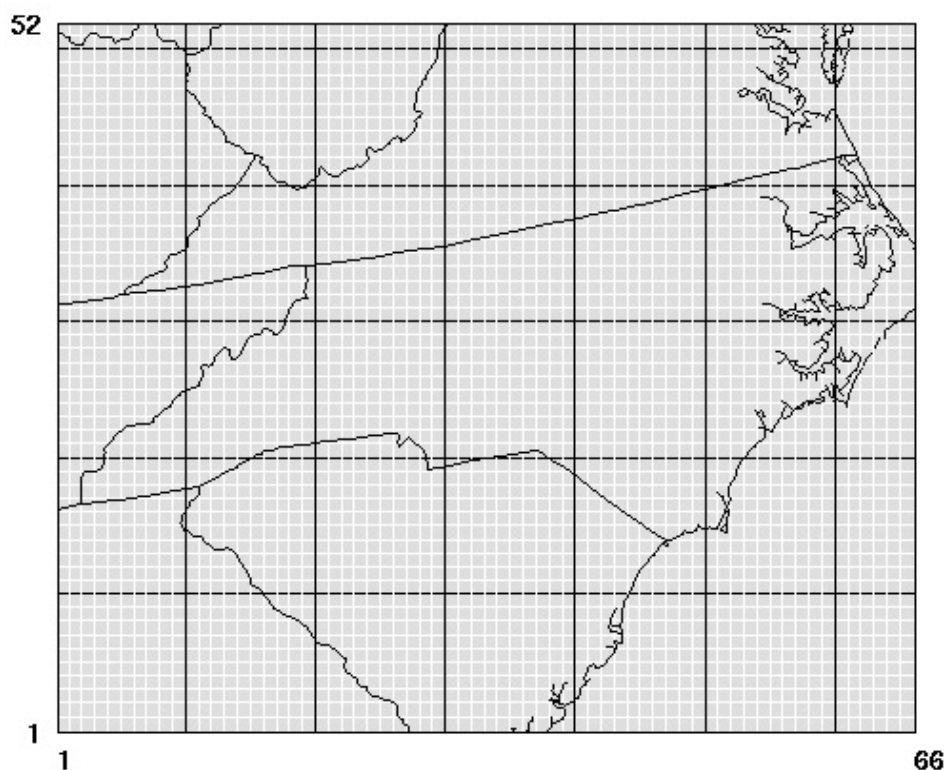
At this time, the case uses North Carolina records *only* from the following inventory data:

stationary area sources: 2005 NEI for CO, NO_x, VOC, SO_2, NH_3, PM_{10}, and $PM_{2.5}$ from nonpt, averfire and alm sectors

nonroad mobile sources: 2005 NEI for CO, NO_x, VOC, SO_2, NH_3, PM_{10}, and $PM_{2.5}$ from nonroad sector

point sources: 2005 NEI for CO, NO_x, VOC, SO_2, NH_3, PM_{10}, and $PM_{2.5}$ from ptipm and ptnonipm sectors

mobile sources: vehicle mileage traveled (VMT) and its average speed (SPEED), and vehicle average hotteling (HOTELLING) for onroad mobile sources, and vehicle population (VPOP) for offroad mobile sources

biogenic sources: BELD3 data processed using BEIS3 (for entire grid, even those cells that are not in North Carolina)

For area, mobile, and point sources, this case uses toxics inventories for ozone modeling. The VOC model species for the current CB5 mechanism are based in part on the toxic inventory. SMOKE resolves the duplicate VOC mass caused by combining VOC emissions with toxic VOC emissions by subtracting the toxic VOC mass from the VOC mass and computing a NONHAPVOC emissions value. This is the "integrate" approach for VOC emissions, described in Section 2.9.5, "Combine toxics and criteria inventories" [55]. The speciation profiles provided with this version of SMOKE include profiles that map the toxics VOC and NONHAPVOC pollutants to current CB5 species. Choices for speciation profiles in this version of SMOKE are listed in Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150].

In addition to the inventory data listed above, the parameters of the case are the following:

- **Episode:** July 10th and 11th of 2005

- **Output time zone:** Greenwich Mean Time (zone 0)

- **Projection:** Lambert Conformal with Alpha=33, Beta=45, Gamma=-97, and center at (-97,40)

- **Domain:** Origin at (1128, -672) kilometers with 66 columns by 52 rows and 12-km square grid cells (shown in Figure 4.1, "nctox domain" [118])

- **Meteorology data:** Daily (25-hour) 2005 meteorology files generated originally by EPA using MM5 and MCIP; the files have been reduced to the modeling domain and contain only the variables needed by SMOKE.

- **Chemical speciation data:** Revised `GSPRO` and `GSREF` files to include toxic pollutants mapped to current-CB5 mechanism for both on-road and nonroad mobile source classification category (SCC) codes.

4.2.2. Script descriptions

4.2.2.1. Overview

Running SMOKE from scripts provides a flexible approach to running SMOKE. We recommend that new users start with the standard configuration of scripts and directories provided with the default installation. This configuration uses the directory structures described in Chapter 3, *SMOKE Directory Structure* [97] and a separate run script for each source category, as we will get to soon. In Section 4.4, "How to use SMOKE" [125], we provide all of the information you need to know to adapt the test case configuration to your modeling case. If you ever want to move beyond using the default configuration, we do not describe how to do that in detail. In fact, we could not describe this, because there are an infinite number of ways to arrange the directories and scripting for SMOKE; however, we have provided some information about what do to in Section 3.3, "How to change the default directory structure" [115].

Running SMOKE using scripts involves many scripts working together. This can be confusing for new users, but we have made efforts to make using the scripts easier and more robust with this version of SMOKE.

All SMOKE scripts can be found in the `$SMKROOT/scripts/run` directory, or simply the `$SCRIPTS/run` directory (after having invoked the Assigns file).

4.2.2.2. Assigns files

The assigns files are located in `$SMKROOT/assigns`. The nctox case assigns file is called `ASSIGNS.nctox.cmaq.cb05_soa.us12-nc`. One Assigns file is provided for the nctox case, and you can create your own files by copying it to a new name and changing it. At least one Assigns file must come with each preconfigured run (such as the nctox case) for SMOKE.

The Assigns files are divided into three major sections. The first section contains environment variables for controlling the setup of a case, such as the dates being modeled, the name of the grid, and the name of the chemical mechanism. The second section lists all of the input files in SMOKE (by source category) and users can change these adapt the Assigns files to their cases by changing the input files. The third section calls helper scripts to create input and output directories and configure the compiler environment variables. It is wise to copy an existing Assigns files to new files and modify the new files, so that you do not lose the default configuration.

In addition, there are a number of helper scripts used by the Assigns files. Unless users are changing the way SMOKE behaves when running from scripts, it should not be necessary to change these scripts. Changing them could result in behavior different from what is documented in this chapter. The helper scripts and their purposes are:

- `check_settings.scr`: Set all environment variables used in Assigns file to default values if they are not already defined by a calling script (e.g., when invoking the Assigns file from the UNIX prompt instead of a script).

- `set_case.scr`: Set the environment variables that control output file names (`ABASE`, `BBASE`, `MBASE`, `PBASE`, `EBASE`, `ASCEN`, `BSCEN`, `MSCEN`, `PSCEN`, and `ESCEN`). See also the `SMK_FUTURE_YN`, `FYEAR`, `SMK_CONTROL_YN`, and `CNTLCASE` script settings in Section 4.2.4, "Script settings" [121]

- `set_dirs.scr`: Set the SMOKE directory structure

- `setmerge_files.scr`: Set **Smkmerge** output file names, depending on run script settings

- `smk_mkdir`: Create input and output directories

- `smk_rmfiles.scr`: Remove SMOKE intermediate and output files

- `sysflags`: Set the compiler options needed for compiling SMOKE

- `unset.scr`: Unset environment variables that do not need to be defined after the Assigns file has been run

4.2.3. Example script files and their purposes

The following files are currently provided with the SMOKE release. There are a number of helper scripts. The only one that users may want to change is the `qa_run.scr` script to add configurations for reporting. The helper scripts are:

- `cntl_run.csh`: Runs the **Cntlmat** and **Grwinven** program. Sets the required `GCNTL` file name depending on script settings, as explained in Section 4.4.11, "Use projection" [155].

- `make_invdir.csh`: Sets the `INVNAME1` and `INVNAME2` settings that **Smkinven** uses for creating the map-formatted inventory intermediate files.

- `metcombine.csh`: Runs the utility program **Metcombine** to combine the `METCRO2D*` and `METCRO3D*` files in the `$METDAT` directory. This script creates files named `METCOMBO*` in the `$METDAT` directory.

- `movelog.csh`: Deletes or renames the log files, depending on the `AUTO_DELETE_LOG` setting.

- `qa_run.csh`: Runs the **Smkreport** program

- `smk_run.csh`: Runs the following SMOKE programs: **Smkinven**, **Rawbio** (for both winter and summer runs), **Normbeis3**, **Spcmat**, **Grdmat**, **Metscan**, **Temporal**, **Tmpbio**, **Tmpbeis3**, **Smkmerge**, **Mrggrid**, and **Smk2emis**.

  Creates `PTMPLIST` file for **Elevpoint** by assuming all files with the name pattern `ptmp*$PSCEN*ncf` in the `$SMKDAT/run_$PSCEN/*` directories should be included in the file.

  Creates the `METLIST` and `RADLIST` files for **Tmpbio** by listing `$MET_FILE1` to `METLIST` and `$MET_FILE2` to `RADLIST`. `MET_FILE1` and `MET_FILE2` are set in the Assigns file.

  Determines which speciation matrix is needed for **Smkmerge**, based on the requested units. If the `MRG_GRDOUT_UNIT` setting contains the string "mole" then the molar speciation matrix and molar gridded, hourly biogenic emissions are set to be used by **Smkmerge**.

  Creates the `FILELIST` file for **Mrggrid** from the `MRGFILES` setting in the calling merge run script.

4.2.4. Script settings

The SMOKE scripts have many environment variables that users can set based on the program documentation in Chapter 6, *SMOKE Core Programs* [211], as well as environment variables that control the scripts. The variables that control the behavior of the scripts are described in this section. The settings are listed in alphabetical order.

Note that using upper case for the names of these settings in the scripts is required. Any values that are Y or N settings must also be upper case.

For documentation on the `A_SPC_OVERRIDE`, `N_SPC_OVERRIDE`, `B_SPC_OVERRIDE`, `M_SPC_OVERRIDE`, and `P_SPC_OVERRIDE` settings, please refer to the `SPC_OVERRIDE` description.

- `AUTO_DELETE`: [Y | N] When set to Y, will delete all SMOKE intermediate and output I/O API files automatically. It will not delete intermediate and output ASCII files or reports.

- `AUTO_DELETE_LOG`: [Y | N] When set to Y, SMOKE log files will be automatically deleted. Otherwise, the files will be automatically renamed by appending a number to the end. The first time the a log file needs to be overwritten, but `AUTO_DELETE_LOG` is N, the file will be renamed to a file name with an "_1" appended to the original name. If the same file needs to be moved on a subsequent rerun and the "_1" file is already there, then a "_2" file will be created.

- `CNTLCASE`: When this is defined to a name of the control case and the `SMK_CONTROL_YN` setting is Y, the following changes are made to the naming environment variables by the `$AS-SIGNS/set_case.scr` file:

```
setenv ASCEN ${ASCEN}_${CNTLCASE}
setenv MSCEN ${MSCEN}_${CNTLCASE}
```

```
setenv PSCEN ${PSCEN}_${CNTLCASE}
setenv ESCEN ${ESCEN}_${CNTLCASE}
setenv FYIOP ${FYIOP}_${CNTLCASE}
```

When the `CNTLCASE` name is set, regardless of the `SMK_CONTROL_YN` setting, the `$AS-SIGNS/set_case.scr` file also resets `FYINV` as follows:

```
setenv FYINV ${FYINV}_${CNTLCASE}
```

This setting is also used for automatic naming of the `GCNTL` file input to **Cntlmat**. See Section 4.4.11, "Use projection" [155].

- `DEBUG_EXE`: Should be set to the name of the fortran debugger (e.g., dbx) to use when running with `DEBUGMODE` set to Y.

- `DEBUGMODE`: [Y | N] When set to Y, the scripts will try to invoke the debug versions of the SMOKE executables. These will only be available to be used if the debug versions of the executables have been created using the **Makeall debug** command in the `$SCRIPTS/make` directory.

- `FYEAR`: The 4-digit future year value is used for renaming files and directories when the `SMK_FU-TURE_YN` setting is Y. When this is true, the following changes are made to the naming environment variables by the `$ASSIGNS/set_case.scr` script:

```
setenv ASCEN    ${ASCEN}-${fyr2}
setenv MSCEN    ${MSCEN}-${fyr2}
setenv PSCEN    ${PSCEN}-${fyr2}
setenv ESCEN    ${ESCEN}-${fyr2}
setenv FYIOP    ${INVOP}-${fyr2}
```

where `${fyr2}` is the last two digits of the `FYEAR` value. These 2 digits are inserted into the output file names for **Smkinven** output files when `SMK_FUTURE_YN` is set to Y.

In addition, when `FYEAR` is defined, the `$ASSIGNS/set_case.scr` script makes the follows settings regardless of the value of `SMK_FUTURE_YN`.

```
setenv FYINV ${FYINV}_${fyr2}
setenv BYFYR ${BYFYR}_${fyr2}
```

This setting is also used for automatic naming of the `GCNTL` file input to **Cntlmat**. See Section 4.4.11, "Use projection" [155].

- `G_STDATE_ADVANCE`: The value should be set to the number of days for 1 run of the **Temporal**, **Laypoint**, and **Smkmerge** programs. The example scripts have this set by the `NDAYS` environment variable in the Assigns file. When this value is set, the Assigns file will automatically increment the `ESDATE` and `G_STDATE` environment variables (that control SMOKE file names and time-dependent programs, respectively) to the next day. This feature is unable to cross from one year to the next.

- `INVTABLE_OVERRIDE`: Sets a file name for the `INVTABLE` file in the Assigns file that overrides whatever value is actually inside the Assigns file. The Assigns file invokes the override mechanism when this environment variable is defined, so to *not* use this scripting feature, the variable must *not* be defined. **In other words, it is not acceptable to set this to a blank value.**

- `MRG_GRDOUT_UNIT`: While this setting is primarily a setting for the **Smkmerge** program, it is also used by the SMOKE scripts to determine whether **Smkmerge** should use mole-based or mass-based speciation matrices and biogenic model-ready files. When the variable is set with the first four letters to "mole", for example moles/hr, mole/s, or moles/day, the **Smkmerge** program will be provided by

the scripts the correct input files. The implementation of this feature currently means that units of "gm mole" will not work properly.

- `NONROAD`: [Y | N] When set to Y, the Assigns file will automatically rename all of the SMOKE area-source intermediate and output files using the "n" prefix on file names instead of the "a" prefix. For example, the `ATMP` file is set to `$SCENARIO/ntmp.ncf` instead of `$SCENARIO/atmp.ncf`. Also, the `AREA` file coming out of **Smkinven** changes to be a different name and use a different map-file directory.

- `PROMPTFLAG`: [Y | N] When set to Y, users will be prompted for all SMOKE file names. This option should not be set to Y for non-interactive processing. In fact, there is little need for this to ever be set to Y.

- `QA_LABEL`: Users can label their **Smkreport** log files and QA reports using this label. Typically, the QA report names are named using the `$FYIOP` variable or `$ESCEN` variable set in the `$AS-SIGNS/set_case.scr` script. When `QA_LABEL` is set to something, the log files and reports are labeled with `$FYIOP.$QA_LABEL` or `$ESCEN.$QA_LABEL`. This renaming is done in the `qa_run.csh` helper script in the `$SCRIPTS/run` directory.

- `QA_TYPE`: [none | all | part1 | part2 | part3 | part4 or custom]. This setting controls the `qa_run.csh` helper script and causes different reports to be written. The `QA_TYPE` settings will cause the script to use different `REPCONFIG` files for input to **Smkreport**, and the `REPCONFIG` files determine which reports are generated. Section 4.4.22, "Change reports" [173] explains these default settings and how to change them to create the reports that you need.

- `RUN_PART1`, `RUN_PART2`, `RUN_PART3`, `RUN_PART4`: These settings should not have to be changed by most users. They are included for the convenience of those who want to more easily rearrange their SMOKE processing. Each source category has its own parts for processing as follows:

	Area	**Biogenic**	**Mobile**	**Point**
`RUN_PART1`	**Smkinven, Spcmat, Grdmat, Cntlmat, Grwinven**	**Rawbio, Normbeis3, Metscan**	**Smkinven, Spcmat, Grdmat, Cntlmat, Grwinven**	**Smkinven, Spcmat, Grdmat, Cntlmat, Grwinven**
`RUN_PART2`	**Temporal**	**Tmpbio, Tmpbeis3**	**Temporal**	**Temporal**
`RUN_PART3`				**Elevpoint**
`RUN_PART4`	**Smkmerge, Mrggrid, Smk2emis**	**Mrggrid, Smk2emis**	**Movesmrg, Mrggrid, Smk2emis**	**Laypoint, Smkmerge, Mrggrid, Smk2emis**

- `RUN_<Program Name>`: [Y | N] The collection of `RUN_<Program Name>` variables control whether a program is going to be run during a given script run. These can be set to N if the program is not needed for some reason (e.g., the REMSAD model does not need the **Laypoint** program run). How these settings should be set is described in Section 4.4.6, "Change which programs are run" [143].

- `SMK_FUTURE_YN`: [Y | N] When set to Y, the file names and output directories are changed to use the `FYEAR` variable setting in the names. See the `FYEAR` documentation for more information on how the naming variables are set. This setting is needed only in scripts for processing future-year cases.

- `SPC_OVERRIDE`: Sets a value for the `SPC` variable in the Assigns file that overrides whatever value is actually inside the Assigns file. The Assigns file invokes the override mechanism when this environ-

ment variable is defined, so to not use this scripting feature, the variable must not be defined. **In other words, it is not acceptable to set this to a blank value.**

In addition, the merge scripts that run the **Mrggrid** program can also use the A_SPC_OVERRIDE, N_SPC_OVERRIDE, B_SPC_OVERRIDE, M_SPC_OVERRIDE, and P_SPC_OVERRIDE settings. These settings override the SPC variable used *only* for the definition of the A_OUT, N_OUT, B_OUT, M_OUT, and P_OUT directories that contain the hourly, gridded, and speciated SMOKE output files. These settings can be used to pull multiple chemical mechanisms (such as toxics and non-toxics) into a single model-ready file. The user must determine which multiple mechanisms are appropriate to combine. It is acceptable to combine toxics-integrated and criteria-only current CB5 mechanisms, but inappropriate to combine CB5 and RADM mechanisms.

- SRCABBR: Sets the source category abbreviation name for use in naming log files. The typical settings are ar (area sources), nr (nonroad sources), mb (mobile sources), bg (biogenic sources), and pt (point sources).

- YEAR_OVERRIDE: Sets a value for the YEAR variable in the Assigns file that overrides whatever value is actually inside the Assigns file. The Assigns file invokes the override mechanism when this environment variable is defined, so to not use this scripting feature, the variable must not be defined. **In other words, it is not acceptable to set this to a blank value.**

4.2.5. SMOKE settings controlled through scripts

All SMOKE programs can be controlled by input environment variables, and the run scripts control these variables. The program documentation in Chapter 5, *SMOKE Utility Tools* [177], Chapter 6, *SMOKE Core Programs* [211], and Chapter 7, *SMOKE Quality Assurance* [305] include the available environment variable settings for each program.

The example run scripts contain most of the major script settings needed for running SMOKE. The major exception to this is the UAM_* settings needed for **Smkmerge** and **Smk2emis**, when running for REMSAD, UAM-V, and CAM$_X$. These can be added to the scripts as needed.

4.3. Running the SMOKE test case

As mentioned in Section 4.1, "Introduction" [117], the first step in working with SMOKE is invoking the Assigns file. In this section, we will go beyond that first step to actually running the scripts for the nctox case. But first, we will review the essential steps for running SMOKE each and every time you start to work with the SMOKE system.

First, the SMK_HOME environment variable needs to be defined. In Section 12.3, "Installing SMOKE" [495], we explain that you can configure your UNIX system to set this automatically each time you log into your computer. If you have not done that, then you will need to set the SMK_HOME variable manually before invoking an Assigns file:

setenv SMK_HOME <directory in which you installed SMOKE>

Second, you must invoke the Assigns file for the case that you want to run. This will make available all of the environment variables that you can use to navigate through the SMOKE directory structure. There is a different Assigns file for each case, so we provide this instruction with the case-specific documentation below. Of course, when you develop an Assigns file for your case, you will use that instead.

NOTE: All of the SMOKE scripts assume that you are running a variant of the C-shell (either **csh** or **tcsh**). The scripts will not work with other shells such as **bash**. If your shell is not **csh** or **tcsh**, you will need to change into an appropriate shell before running any SMOKE scripts.

To run the nctox test case, you must first source the Assigns file using the commands below:

cd $SMK_HOME/subsys/smoke/assigns *(change to the assigns directory)*

source ASSIGNS.nctox.cmaq.cb05_soa.us12-nc *(set up for the nctox case)*

Next, you'll run the base-year nctox scripts. These scripts process the stationary area, biogenic, nonroad mobile, and point emissions for the base year. The commands to run the scripts are below:

cd $SCRIPTS/run *(change to the run scripts directory)*

smk_area_nctox.csh *(invoke the stationary area run script)*

smk_bg_nctox.csh *(invoke the BEIS3 biogenic run script)*

smk_nonroad_nctox.csh *(invoke the nonroad mobile run script)*

smk_point_nctox.csh *(invoke the point run script)*

smk_rateperdistance_nctox.csh *(invoke the MOVES mobile sources on-roadway rate-per-distance (RPD) run script for all processes)*

smk_ratepervehicle_nctox.csh *(invoke the MOVES mobile sources off-network rate-per-vehicle (RPV) run script)*

smk_rateperprofile_nctox.csh *(invoke the MOVES mobile sources off-network rate-per-profile (RPP) run script)*

smk_rateperhour_nctox.csh *(invoke the MOVES mobile sources off-network rate-per-hour (RPH) run script)*

smk_mrgall_nctox.csh *(invoke the all-sources merge script)*

To verify that the nctox scripts have run correctly, go to the log file directory and look for errors by using:

cd $LOGS *(change to the log file directory for the test case)*

```
grep ERROR *
```

If there are no errors, the next step is to run a QA script to be sure that the answers match.

cd $SCRIPTS/install *(change to the install directory)*

check_smk_install *(invoke the smoke install quality assurance script)*

4.4. How to use SMOKE

This section builds on the previous section to describe how to use the SMOKE files, scripts, and programs to create emission inputs for your air quality model. The first part of setting up SMOKE for a case other than the test case that comes with SMOKE is to create an Assigns file and script files for your case. If you have not read through the earlier parts of this chapter, we recommend that you do so. There are some critical points that you need to understand about the Assigns file and scripts. In Section 4.1, "Introduction" [117], we have described that an Assigns file must always be used when starting to work with SMOKE and the UNIX commands to do that. In Section 4.2.2, "Script descriptions" [119], we have already described the

purpose and use of the Assigns file and scripts, so please refer to that section for more information on these.

You may use this section as a step-by-step guide for all of the changes that you need to make to the scripts and files for running SMOKE. If you read it straight through and follow all of the steps that apply to your situation, then you will have set up SMOKE for your modeling case. If you are already familiar with SMOKE, you may also use this section as a reference for changing specific parts of the configuration to meet specific needs.

To create your own scripts and use your own data for the first time, you must take the following steps:

1. Copy the example Assigns file to your own copy. The name of the Assigns file is not critical to the success of this step, but it is useful to name the file something that you will be able to identify easily for the modeling case for which it will be used.

   `cp ASSIGNS.nctox.cmaq.cb05_soa.us12-nc <yourfile>`

2. Modify the necessary features of the Assigns file for your case. For example, change the scenario names, episode dates, duration, grid name, or speciation name. Some of these changes require additional input files. See Section 4.4.1, "Change Assigns file to set scenario names, grid names, and other case-specific configuration information" [126] for more details on the settings that you can change and the values to use for each.

3. Invoke the Assigns file. This will create all of the input and output directories that you will need to set up your case:

   `source <yourfile>`

4. Copy the appropriate nctox scripts to your own scripts, change the Assigns file listed in the script to the one that you created in the previous step.

5. Using one or more of the subsections in this chapter, create your input files and change the options in the scripts. The subsections below explain what files and settings need to be changed and checked for many types of operations for configuring SMOKE.

You need not always start from the example Assigns file. Once you have your own case-specific Assigns files, you may use them as starting points for creating other files.

4.4.1. Change Assigns file to set scenario names, grid names, and other case-specific configuration information

This subsection provides an overview of all of the Assigns file settings that you may need to change. The remaining subsections in this chapter include more detail about the changes as associated with specific modeling needs. For example, Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150] explains how to change the Assigns file to set different chemical mechanisms as well as all of the files needed.

In the 5-step list provided in the previous section, note that the Assigns file must first be changed (step 2) and then invoked (step 3) to create the input and output directories for your case. Ideally, all of the settings in the list below would be done in step 2 before step 3 is performed. If this is not possible, however, you should *at a minimum* set the first five items on the list below before moving to step 3. These are the variables that affect the input and output directory names, so if they are set, then the directories that you need to use will for all other configuration steps will be set up for you.

The list below provides the typical settings that may need to be changed to be consistent with the case that you need. Each item in the list references a different section of this chapter and Table 4.1, "Variable in Assigns file for naming files and directories" [128] or Table 4.2, "Variables in Assigns file for controlling time period and grid of processing" [130] for more information. The variables in the Table 4.1, "Variable in Assigns file for naming files and directories" [128] are used for naming files and directories. Table 4.2, "Variables in Assigns file for controlling time period and grid of processing" [130] includes the settings for controlling the time period and grid of your SMOKE runs.

The checklist of major Assigns settings to change is:

1. Change the `INVID`, `INVEN`, and `INVOP` values to a new inventory name. The simplest approach is to set these three variables to the same name, but different names can be used. Examples of which files or directories are affected by these settings are provided in Table 4.1, "Variable in Assigns file for naming files and directories" [128]. See also Section 4.4.2, "Use new inventory data" [131].

2. Change the `ABASE`, `BBASE`, `MBASE`, `PBASE`, and `EBASE` values to a new scenario name. The simplest approach is to set these three variables to the same name, but different names can be used. Examples of which files or directories are affected by these settings are provided in Table 4.1, "Variable in Assigns file for naming files and directories" [128]. See also Section 4.4.2, "Use new inventory data" [131].

3. Change the `METSCEN` value to your meteorology scenario name. This will affect the name of the input directory for the meteorology files (the `$METDAT` directory, which can be a linked directory if you do not want to install your meteorology files under the SMOKE directory structure).

4. Change the `SPC` value to one of the supported speciation mechanisms (see Section 4.4.10.1, "Choosing from default mechanisms" [150]) or to a new speciation mechanism (see Section 4.4.10.2, "Setup of new mechanism or changing an existing mechanism" [153]).

5. Change the `GRID` and `IOAPI_GRIDNAME_1` values to the grid names for naming files. See also Section 4.4.9, "Use a new modeling grid or change spatial inputs (anthropogenic and biogenic)" [146].

6. Change the run period naming settings `ESDATE` and `NDAYS` (see Table 4.1, "Variable in Assigns file for naming files and directories" [128]). See also Section 4.4.13.4, "Changing the dates, times, and duration of model-ready SMOKE emissions files" [163].

7. Change the episode settings `EPI_STDATE`, `EPI_STTIME`, `EPI_RUNLEN`, and `EPI_NDAY` and the run period settings `G_STDATE`, `G_STTIME`, and `G_RUNLEN`. See Table 4.2, "Variables in Assigns file for controlling time period and grid of processing" [130] and Section 4.4.13.3, "Changing the episode" [162] and Section 4.4.13.4, "Changing the dates, times, and duration of model-ready SMOKE emissions files" [163].

8. Change the `YEAR` value to the year of your base case inventory.

Table 4.1. Variable in Assigns file for naming files and directories

Environment Variable	Description
INVID	Inventory input directory identifier. Controls directory path of input inventory. INVDIR = $SMKDAT/inventory/$INVID
INVOP	Base year inventory output name. Controls output file names for SMOKE intermediate inventory files and other output files from **Smkinven** and inventory-only reports from **Smkreport**. This variable is used to initialize the FYIOP variable, which is also used in the Assigns file and is equal to INVOP for base case modeling. FYIOP is reset by the set_case.scr helper script (see Section 4.2.4, "Script settings" [121] documentation on FYEAR and CNTLCASE)
INVEN	Base year inventory name, with version. Controls directory path of output inventory. INVOPD = $SMKDAT/inventory/$INVEN
ABASE	Scenario identifier for area/non-point and nonroad mobile sources. Controls output file names and directories for outputs from SMOKE programs other than **Smkinven** (e.g., locations and file names of all matrices, hourly emissions, model-ready files). This variable is used to initialize the ASCEN variable, which is also used in the Assigns file and is equal to ABASE for base case modeling. ASCEN is reset by the set_case.scr helper script (see Section 4.2.4, "Script settings" [121] documentation on FYEAR and CNTLCASE). Should be set to the same value as BBASE, MBASE, PBASE, and EBASE for the basic configuration.
BBASE	Scenario identifier for biogenic sources that works the same way as ABASE, but affects the BSCEN variable.
MBASE	Scenario identifier for on-road mobile sources that works the same way as ABASE, but affects the MSCEN variable.
PBASE	Scenario identifier for on-road mobile sources that works the same way as ABASE, but affects the PSCEN variable.
EBASE	Scenario identifier for merged emissions sources. Controls output file names and directories for outputs for final model-ready files. Sets the path for the STATIC, SCENARIO, and OUTPUT directories and file names for the merged outputs for multiple source categories from the **Smkmerge** program (e.g., the EGTS3D_L file).
METSCEN	Meteorology scenario name. Sets the METDAT directory for the MCIP meteorology files and can be used to set the meteorology file names (not done in example nctox scripts). Also helps set names for meteorology-dependent SMOKE intermediate files for biogenics (**Tmpbeis3** outputs B3GTS_S and B3GTS_L), on-road mobile (**Temporal** output MTMP), and point (**Laypoint** outputs PLAY and PLAY_EX).
GRID	Grid name used for naming files that depend on the selected grid. User must coordinate GRID settings with IOAPI_GRIDNAME_1 setting Table 4.2, "Variables in Assigns file for controlling time period and grid of processing" [130]. Dependent files include gridding matrix, gridded land use and biogenic emissions, hourly mobile emissions, layer-fractions files, model-ready emissions, and some reports.
SPC	Speciation profile name for selecting input files and naming output files and directories that depend on the chemical mechanism. This setting can be overridden by the SPC_OVERRIDE script setting (see Section 4.2.4, "Script settings" [121]). The SPC setting sets input files GSPRO and GSREF; speciation matrix intermediate file names, and output directories OUTPUT, A_OUT, B_OUT, M_OUT, N_OUT, and P_OUT.

Environment Variable	Description
ESDATE	Starting date for first modeled period in episode. Has format YYYYMMDD where YYYY is the 4-digit year, MM is the two-digit month (e.g., 07 for July), and DD is the two-digit day (e.g., 01 for the first). Users should coordinate this with the `G_STDATE` setting Table 4.2, "Variables in Assigns file for controlling time period and grid of processing" [130]. The Assigns file resets this variable if the `G_STDATE_ADVANCE` variable is set by a calling script (see Section 4.2.4, "Script settings" [121]).
MSDATE	Meteorology file dates. This variable can be used to help name time-dependent meteorology files or directory names. In the example scripts for the nctox case, however, this variable is not used and the `G_STDATE` variable is used instead.
NDAYS	Number of days in a modeling period. To break up a modeling episode into multiple modeling periods of one or more days, the user must set and coordinate this setting with the `G_RUNLEN` setting Table 4.2, "Variables in Assigns file for controlling time period and grid of processing" [130]. `NDAYS` represents the number of days indicated in hours by the `G_RUNLEN` variable. `NDAYS` is used for naming hourly emissions files output from the **Temporal** program, biogenic outputs from **Tmpbio**, layer fractions files from **Laypoint**, and model-ready emissions files from **Smkmerge**.
YEAR	Base case modeling year. This setting is used to set several things in coordination with the `FYEAR` and `SMK_FUTURE_YN` script settings (see Section 4.2.4, "Script settings" [121]). For growth modeling, it is used in the file name for the `GCNTL` input file, set by the growth/control example nctox scripts. This setting can be overridden by the `YEAR_OVERRIDE` setting, also described in Section 4.2.4, "Script settings" [121]. Lastly, it is used to set the file name for the winter/summer designations file output from **Metscan**, called the `BIOSEASON` file.

Table 4.2. Variables in Assigns file for controlling time period and grid of processing

Environment Variable	Description
`G_STDATE`	Julian starting date of first modeled period in episode. Has format YYYYDDD where YYYY is a 4-digit year and DDD is a 3-digit Julian date (e.g., 001 for the first day of the year). Users should coordinate this with the `ESDATE` setting Table 4.1, "Variable in Assigns file for naming files and directories" [128]. The Assigns file resets this variable if the `G_STDATE_ADVANCE` variable is set by a calling script (see Section 4.2.4, "Script settings" [121]). Controls operation of time-dependent programs **Temporal**, **Tmpbio**, **Laypoint**, and **Smkmerge**.
`G_STTIME`	Start time of each period in the episode. Has format HHMMSS where HH is a 2-digit hour (leading zero not required), MM is a 2-digit minute (set to 00 for SMOKE), and SS is a 2-digit second (set to 00 for SMOKE).
`G_RUNLEN`	Duration of each run period in episode. To break up a modeling episode into multiple modeling periods of one or more days, the user must set and coordinate this setting with the `NDAYS` setting Table 4.1, "Variable in Assigns file for naming files and directories" [128]. `G_RUNLEN` has format HHHHMMSS, where HHHH is a 4-digit duration in hours (leading zeros not required), MM and HH should be set as for `G_STTIME`. `G_RUNLEN` controls the duration modeled for each run of the **Temporal**, **Tmpbio**, **Laypoint**, and **Smkmerge** programs. Must be set one hour greater than the number of days times 24 (e.g., a two-day period length is 2 x 24 + 1 = 49).
`G_TSTEP`	Time step. Has format HHMMSS with format the same as described for `G_STTIME`. The current version of SMOKE allows this setting to only be 10000, for hourly timesteps.
`EPI_STDATE`	Episode start date. Has the same format as `G_STDATE`. Does not change during the course of an episodic run. In the Assigns file, should be set the same as `G_STDATE`.
`EPI_STTIME`	Episode start time. Has the same format as `G_STTIME`. Does not change during the course of an episodic run. In the Assigns file, should be set the same as `G_STTIME`.
`EPI_RUNLEN`	Episode duration. Has the same format as `G_RUNLEN`. Does not change during the course of an episodic run. In the Assigns file, should be set for the entire duration of the modeling episode, which must be greater than or equal to `G_RUNLEN` minus one hour, is limited to at most one year, and for best results, should be a multiple of `G_RUNLEN` minus one hour. For example, if `G_RUNLEN` is 250000 (one day plus the required one additional hour), `EPI_RUNLEN` must be at least 240000 and at most 365 x 240000 = 87600000. This example would run an annual episode using 24-hour run periods.
`EPI_NDAY`	The number of days in the episode. Should be the value of `EPI_RUNLEN` divided by 240000. This settings controls for how many days to use the automatic looping in the example scripts, which loops over all of the periods (set by `G_RUNLEN`) during the episode (set by `EPI_RUNLEN`).
`IOAPI_GRIDNAME_1`	The selected grid. Controls which grid will be selected by the **Grdmat** program from the `GRIDDESC` file. Users should coordinate this setting with the `GRID` setting from Table 4.1, "Variable in Assigns file for naming files and directories" [128], which is used for naming files. This setting must match the grid names in the gridded land use and spatial surrogates files' headers.

4.4.2. Use new inventory data

This section describes what steps are necessary to take when using new inventory data for anthropogenic sources. For information on biogenic sources, see Section 4.4.15, "Use BEIS3" [166]. For more information about the details of each source category (e.g., what is the difference between using MOVES emission factors or precomputed on-road mobile emissions), see Section 2.8.1, "Summary of SMOKE processing categories" [29].

Because this section is so spread out, we provide a small section guide here. This section includes the following subsections:

- Area/non-point inventories
- Nonroad mobile inventories
- On-road mobile inventories
- Point inventories
- Day-specific and hour-specific point and fire inventories
- Future-year inventories
- Pre-gridded inventories (for same output modeling domain or for converting to a different modeling domain (example: EDGAR to CMAQ ready emissions file)
- Using seasonal and annual data in the same run
- Files and settings for all inventories

4.4.2.1. Area/non-point inventories

1. Determine the file format that you have (for preformatted data) or want to create (for new data).

   a. Preformatted data considerations

      - If you have preexisting data in SMOKE input format, then you will most likely want to use that format to input to SMOKE, unless you discover that you need a different format to take advantage of a SMOKE feature not available in a different format.

      - If you have multiple files in a single format, then you can combine these into a single SMOKE run using a list file.

      - If you have data in more than one format other than the case just mentioned, you must convert all files to the same format or process them separately in SMOKE, merging the hourly, gridded, speciated emissions files using **Mrggrid**.

      - If your area inventory includes nonroad mobile sources, you can process nonroad mobile sources with stationary area sources in most cases (see next bullet for exception).

      - If you want to combine nonroad mobile criteria and toxics inventories, you need to process nonroad mobile sources separately from area sources (see Section 4.4.2.2, "Nonroad mobile inventories" [133]).

      - Pregridded data from any source category can be input to SMOKE as an area source, but you must convert the data to an I/O API NetCDF file format first. See Section 4.4.2.7.1, "Pregridded inventories from the same modeling domain as an area source without source characterization or attributes" [138] for more information.

2. Collect or create inventories in correct formats.

Inventory files for area sources are usually named starting with "arinv", because ARINV is the logical file name (environment variable) that **Smkinven** uses to read the file. The formats available for area/non-point sources are:

a. List format. The list format is simply an ASCII file that lists all of the files that you want SMOKE to import in a single run.

b. List Grid format. The list grid format is simply an ASCII file that lists all of the global gridded inventory files that you want SMOKE to import in a single run. The List Grid format has four columns. Assigned SCC, Pollutant ID, Local Variable Name, Month of inventory and Full inventory fine directory path and filename.

c. ORL and FF10 formats. These formats are most appropriate for non-point toxics inventories, but also can be used for any inventory data.

d. Pregridded I/O API format. This format can only be used for pregridded data that has been converted to a time-independent, gridded I/O API file. The variables in the file are the pollutants. No SCCs or other source information can be provided with this format.

3. Set and/or check the file headers that apply for your file format.

a. File format header: Make sure that the first line of the list file or individual files start with the #LIST (for list files only),#LIST GRID, #ORL, #FF10 (see Section 8.2.1.2, "File format header" [351]).

b. File type (#TYPE header): The type is important because it helps SMOKE determine the file format. Make sure that the header has been set according to the directions in Section 8.2.1.4, "Header records" [352].

c. Country (#COUNTRY header): Make sure that this head has been used and that the country name used in this header matches the name in the country section of the COSTCY or GEO-CODE_LEVEL[1-4] file (if USE_EXP_GEOCODE Y) file.

d. Inventory year (#YEAR). Make sure that the inventory year is accurately reflected in the header. This is the only way SMOKE knows what year the data is for. If this year is not a base year but a future-year, make sure to follow the additional instructions in Section 4.4.2.6, "Future-year inventories" [137].

e. Description (#DESC header): Although this header is not required, it is good practice to take advantage of it and include as much information about the data as you have. There is no limit to the number of #DESC headers that can be included in the file.

f. Data header (#POLID or #DATA header): Make sure that this required header is included in the file. It should accurately reflect the data values provided on each line of the file. If there are data fields at the end of the file that you want to drop from being imported, you can leave them off of the #POLID or #DATA header to cause them to not be imported by **Smkinven** (see also the INVTABLE file for dropping emissions during import).

For ORL format, this header does not apply. The pollutants are designated by code number (i.e., CAS number) in the file format and therefore the header is not needed.

4. Install inventory files in $ARDAT directory.

This directory is $SMKDAT/inventory/$INVID/area.

5. Ensure the Assigns file has the `ARINV` variable set to the `ARINV` file name for your area-source inventory file.

6. Optionally, set up to assign point-source locations to area sources using the `ARTOPNT` file.

    a. Put the `ARTOPNT` file in the `$INVDIR/other` directory.

    b. Ensure the name of the `ARTOPNT` file is consistent with the `ARTOPNT` setting in the Assigns file.

    c. Make sure that the `ARTOPNT` file data are consistent with the year of your inventory (e.g., use 1999 coordinates for a 1999 inventory), or that you can accept any discrepancies.

7. See also steps for all source categories in Section 4.4.2.8, "Files and settings for all inventories" [140].

4.4.2.2. Nonroad mobile inventories

1. Determine the file format that you have (for preformatted data) or want to create (for new data).

    a. • See Section 4.4.2.1, "Area/non-point inventories" [131] for issues relating to stationary area / non-point sources. These are the same issues as for nonroad mobile sources.

       • If you need to process nonroad mobile separately from stationary area/non-point sources because of recommendations in Section 4.4.2.1, "Area/non-point inventories" [131], you must separate the stationary area/non-point and nonroad mobile sources to two separate files if they are not already separated. There is no SMOKE utility to do this, but the UNIX **grep** command can be used to search for all SCCs in the nonroad mobile inventory; for example, all SCCs starting with "22".

    b. New data considerations

       • See Section 4.4.2.1, "Area/non-point inventories" [131] for issues relating to stationary area and non-point sources. These are the same issues as for nonroad mobile sources.

       • Decide whether area and nonroad mobile sources will be processed together, which may depend on whether the available inventory data include both area and nonroad mobile or whether they are already separate. One reason for separate processing is if you anticipate control strategies that affect only stationary area or only nonroad mobile sources, though it is not required that the processing be separate to apply such controls. A second reason for separate files if you are also using toxics inventories.

    If processing nonroad mobile sources separately, continue with these steps. Otherwise, the steps for area/non-point sources listed in Section 4.4.2.1, "Area/non-point inventories" [131] will be sufficient.

2. Collect or create inventories in correct formats.

    Inventory files for nonroad sources are usually named starting with "nrinv", because the Assigns file uses the `NRINV` environment variable to set the `ARINV` logical file name (environment variable) that **Smkinven** uses to read the file. The list of available formats for nonroad mobile sources is:

    a. List format. The list format is simply an ASCII file that lists all of the files that you want SMOKE to import in a single run.

    b. ORL format. This format is most appropriate for criteria and toxics inventories, but also can be used for any inventory data.

c. FF10 format. This format is most appropriate for criteria and toxics inventories, but also can be used for any inventory data.

3. Set and/or check the file headers that apply for your file format.

   a. File format header: Make sure that the first line of the list file or individual files start with the #LIST (for list files only),#LIST GRID, #ORL, #FF10.

   b. For other headers, please refer to the documentation in step (3) of the area/non-point steps in Section 4.4.2.1, "Area/non-point inventories" [131].

4. Install inventory files in $NRDAT directory.

   This directory is $SMKDAT/inventory/$INVID/nonroad.

5. Ensure the Assigns file has the NRINV variable set to the NRINV file name for your new nonroad-mobile-source inventory file.

6. Optionally, set up to assign point-source locations to nonroad mobile sources using the ARTOPNT file.

   Please refer to the documentation in step (6) of the area/non-point steps in Section 4.4.2.1, "Area/non-point inventories" [131].

7. See also steps for all source categories in Section 4.4.2.8, "Files and settings for all inventories" [140].

4.4.2.3. On-road mobile inventories

1. Determine the file format that you have (for preformatted data) or want to create (for new data).

   • If using MOVES emission factors through SMOKE, create a VMT file in FF10 on-road mobile activity format (or use in combination with list format).

   • If using both VMT and emissions, you *must* use a list-format to import the two separate inventories together.

   • If using both criteria and toxics data and wish to have the full functionality of SMOKE for both inventories, you must use a list file to list both inventory files (criteria and toxics) for SMOKE to combine.

2. Collect or create inventories in correct formats.

   Inventory files for on-road mobile sources are usually named starting with "mbinv", because **Smkinven** uses the MBINV logical file name (environment variable) to read the file. The list of available formats for on-road mobile sources is:

   a. List format. The list format is simply an ASCII file that lists all of the files that you want SMOKE to import in a single run.

   b. ORL and FF10 format. These formats are most appropriate for toxics inventories, but also can be used for any inventory data.

3. Set and/or check the file headers that apply for your file format.

   a. Please refer to the documentation in step (3) of the area/non-point steps in Section 4.4.2.1, "Area/non-point inventories" [131].

4. Install inventory files in `$MBDAT` directory.

   This directory is `$SMKDAT/inventory/$INVID/mobile`

5. Install other mobile-specific files in inventory directories.

   The following mobile-specific files must also be installed in the `MBDAT` directory. These files can usually be copied from the example files provided with the SMOKE installation. The formats and additional discussion about the contents of these files can be found in Section 8.9.2, "Mobile" [424]. These files are:

   • `MCXREF` (MOVES usage only)

   • `MFMREF` (MOVES usage only)

   • `MEPROC` (MOVES usage only)

   For all inputs needed for MOVES usage only, refer to Section 4.4.12, "Use MOVES for on-road mobile sources" [159].

6. See also steps for all source categories in Section 4.4.2.8, "Files and settings for all inventories" [140].

4.4.2.4. Point/fires inventories

1. Determine the file format that you have (for preformatted data) or want to create (for new data).

   a. Preformatted data considerations

      • If you have preexisting data in SMOKE input format, then you will most likely want to use that format to input to SMOKE, unless you discover that you need a different format to take advantage of a SMOKE feature not available in a different format.

      • Pregridded data must be input as an area source.

   b. New data considerations

      • Hour-specific or day-specific data (including precomputed plume-rise data) are covered in Section 4.4.2.5, "Day-specific and hour-specific point and fire inventories" [136].

      • For wildfires and prescribed fires emission modeling without precomputed plume rise, you could use `ORL FIRE` format for both criteria and toxics inventory together. Using a master (annual) inventory file (`PTINV`) for fires described in Section 8.2.8.3, "ORL FIRE Format" [380], SMOKE requires day-specific inventory file(s) (`PTDAY`) described in Section 8.2.6.2, "ORL FIRE Emission Format" [368] to provide both criteria and toxics inventories for those matched sources. See more infomation about how-to-do at Section 4.4.17, "Plume Rise Calculation for Fires" [167].

2. Collect or create inventories in correct formats.

   a. List format. The list format is simply an ASCII file that lists all of the files that you want SMOKE to import in a single run.

   b. ORL format. This format is most appropriate for point toxics inventories, but also can be used for any inventory data.

3. Set and/or check the file headers that apply for your file format.

Please refer to the documentation in step (3) of the area/non-point steps in Section 4.4.2.1, "Area/non-point inventories" [131].

4. Install inventory files in `$PTDAT` directory.

    This directory is `$SMKDAT/inventory/$INVID/point`

5. Ensure the Assigns file has the `PTINV` variable set to the `PTINV` file name for your point-source inventory file.

6. For hour-specific and day-specific data, including fire data, see Section 4.4.2.5, "Day-specific and hour-specific point and fire inventories" [136].

7. See also steps for all source categories in Section 4.4.2.8, "Files and settings for all inventories" [140].

4.4.2.5. Day-specific and hour-specific point and fire inventories

Point sources in SMOKE can include day-specific or hour-specific data. This can include fire emissions and plume rise data. The following steps must be performed to use day-specific and/or hour-specific data in SMOKE.

1. Determine the files that need to be created and their formats

    a. For day-specific emissions data, create the list file `PTDAY` and one or more files with day-specific emissions in the format provided in Section 8.2.6, "PTDAY: Point source day-specific emissions" [367]. The format requires that a list file be used to point to the actual data files, even if a single data file with emissions data is used.

    b. For day-specific emissions data, the daily total should ideally be based on the same time zone as is being modeled by SMOKE. SMOKE needs to use the same output time zone (set with the `OUTZONE` setting) as is used in the meteorology data that will be input to the air quality model. For example, if the time zone being modeled by SMOKE were GMT, then ideally the daily total emissions would be from 0 GMT to 23 GMT for each day.

    c. For hour-specific emissions data that is *not* CEM data, create the list file `PTHOUR` and one or more files with hour-specific emissions in the format provided in Section 8.2.7, "PTHOUR: Point source hour-specific emissions" [369]. The format requires that a list file be used to point to the actual data files, even if a single data file with emissions data is used.

    d. If using CEM data, the data files for 1995 through 2000 are available with SMOKE. A list file `PTHOUR` must be created to list the one or more CEM data files that will be input. To speed processing, the DATERANGE setting can be used in the list file `PTHOUR` to cause SMOKE to skip the dates in the CEM files that are not needed. Detail information is described in Section 2.9.19, "Processing hour-specific CEM data" [62]

    e. If using hourly precomputed plume rise (e.g., for fire data), the `PTHOUR` file should be created as described above for emissions data, but specific variable names must be provided. The allowable data includes the fraction of the plume in layer 1, the plume bottom height, and the plume top height.

    f. If using daily fire inventories without precomputed plume rise, the `PTDAY` file should be created as described above for emissions data and specific variables (size of acres burned, fuel loading, and duration of fire) must be provided for plume rise calculation and re-normalization of temporal profiles, as described in the file format documentation in Section 8.2.8.3, "ORL FIRE Format" [380] and Section 8.2.6.2, "ORL FIRE Emission Format" [368].

g. If using hourly stack exit temperature, velocity, or flow rates, the PTHOUR file should be created as described above for emissions data, but specific variable names must be provided, as described in the file format documentation in Section 8.2.7, "PTHOUR: Point source hour-specific emissions" [369].

2. Install inventory files in `$PTDAT` directory.

   This directory is `$SMKDAT/inventory/$INVID/point`

3. Ensure the Assigns file has the PTHOUR and PTDAY variables set to the PTHOUR and PTDAY file names for your inventory files.

4. Ensure script settings will use the day-specific and hour-specific data.

   a. For day-specific data import and usage, set DAY_SPECIFIC_YN setting to Y.

   b. For hour-specific data import and usage, set HOUR_SPECIFIC_YN setting to Y. This applies to using either hourly emissions or hourly-precomputed plume rise.

5. If using hourly data, make sure the OUTZONE setting is consistent with what will be used for other SMOKE processing steps.

   The example SMOKE scripts make it difficult to not use the same OUTZONE setting for all processing steps. Nevertheless, the user is responsible for ensuring that the same setting is used for all SMOKE processing steps, including import of hourly inventories. The OUTZONE setting will affect the hourly inventory import because the hourly emissions will be adjusted by **Smkinven** from the local time zones in the PTHOUR input files to the output time zone set by OUTZONE to create the PHOUR intermediate file.

6. If using precomputed plume rise, ensure script settings will cause SMOKE to use the plume rise data.

   a. Set HOUR_PLUMEDATA_YN setting to Y.

   b. For UAM, REMSAD, and CAM_X modeling, set:

      • RUN_LAYPOINT setting to Y (this program is not usually needed for these models)

      • EXPLICIT_PLUMES_YN setting to Y

7. For fires emission modeling with internal plume rise calculation, SMOKE compute the plume rise data with internally computed heat flux from fuel loading and burned area. See detail at Section 4.4.17, "Plume Rise Calculation for Fires" [167]

8. If providing future-year hour-specific data, ensure that the dates used in the file are consistent with the *base year* dates. The dates must match the dates of the meteorology data that will be used by the air quality model.

4.4.2.6. Future-year inventories

For future-year modeling, you can choose to input future-year inventories to SMOKE. This is a different approach from importing a base-year inventory and using SMOKE growth to create a future-year inventory. The future-year inventory could be created outside of SMOKE and would need to be in one of the SMOKE input formats available for the source categories of interest.

A future-year inventory could also be created by SMOKE, since the **Grwinven** program can output emissions in ORL format. In this section, we do not cover importing base year inventories and using SMOKE to

grow them to a future year; that topic is covered in Section 4.4.11, "Use projection" [155]. Following the instructions in that section, it is feasible to create a future-year inventory in SMOKE raw input format and use it as we describe in this section.

1. Set the #YEAR setting in the inventory input files to the future inventory year.

2. Set the SMK_BASEYR_OVERRIDE setting in the run script to the value of the base year.

   Ordinarily, this setting would simply be set to zero, which indicates that it is not to be used and the inventory year is the same as the base year. In this case, the base year cannot be set by the inventory year, so this setting provides the actual year to use. The base year is most often the same year as the year of meteorology data you are using to model your episode. When this setting is defined by a script, **Smkinven** will set the base year with that setting and set the future year as the year of the inventory data.

3. Set the YEAR setting in the Assigns file to the base year.

4. Set the FYEAR setting in the run scripts to the future year.

5. Set the SMK_FUTURE_YN setting in the run scripts to Y.

See Section 4.2.4, "Script settings" [121] for more information about how this setting affects the output directory structure and file names.

4.4.2.7. Pre-gridded inventories

Smoke has two ways of processing the pregridded inventory data. If the inventory data is in native netCDF format then see Section 4.4.2.7.1, "Pre-gridded inventories from the same modeling domain as an area source without source characterization or attributes" [138]. If the inventory is in IO-API/netCDF format then see Section 4.4.2.7.2, "Pre-gridded inventories processing for converting between different modeling domains with source characterization and attributes" [139]

4.4.2.7.1. Pre-gridded inventories from the same modeling domain as an area source without source characterization or attributes

SMOKE has a limited capability to import pre-gridded data from the same modeling domain. It can import the data if it is time-independent and if the grid cell and pollutant names are the only attributes needed to identify the emissions. If SCC or other attributes are available, SMOKE cannot use these directly; though in the case of SCCs, each SCC could conceivably be tediously input as a separate dataset and modeled as a separate inventory.

If any source category is available as pre-gridded data, the following steps must be taken.

1. Plan to import the data as an area source.

   Only SMOKE area-source processing can use pre-gridded data, no matter that actual source category of origin for the data. Essentially, once emissions data no longer has attributes such as SCCs, stack information for point sources, or country/state/county codes, it all appears to be the same type of data to SMOKE and can be processed as an area source.

   This is not to say that all pre-gridded inventories can be processed correctly by SMOKE. For example, if a pre-gridded point source inventory including stack identifiers and stack parameters were provided with grid cell numbers instead of lat/lon coordinates, this could not be appropriately processed by SMOKE area processing as pre-gridded data.

If any SMOKE processing steps such as chemical speciation or temporal allocation need to be done based on SCC, this is not possible. Only a single default profile for speciation or temporal allocation can be applied to each pre-gridded inventory.

2. Convert data to I/O API format.

    Users must convert the pre-gridded data to I/O API format. The format is like the AG file that can be output from **Smkmerge**. This format has each pollutant stored as a variable and is time independent. Some sort of ASCII-to-I/O API converter must be created and used to perform this step. At this time, there is no general-purpose tool to make such a conversion released with the SMOKE system.

3. Set the IMPORT_GRDIOAPI_YN setting to Y.

    This setting will cause **Smkinven** to expect the I/O API gridded input and will request the AG logical file name instead of the ARINV file.

4. Ensure that the AG logical file name is set by the Assigns file or by the run script.

    This logical file name must be set to the I/O API-formatted pre-gridded inventory file.

4.4.2.7.2. Pre-gridded inventories processing for converting between different modeling domains with source characterization and attributes

SMOKE will regrid the data from the native NetCDF-formatted pregridded inventory data (e.g. lat-lon for global inventories) to the CMAQ modeling grid, and apply a unique source ID to each grid cell to allow SMOKE to perform chemical speciation, temporal allocation from annual to hourly timesteps, and elevated source selection and vertical allocation. The pre-gridded inventory (EDGAR) files each contain data for a different sector, the sector and pollutant contained in each pre-gridded inventory file is specified in the ARINV file using the #LIST GRID format.

To process pre-gridded data from different modeling domains, the following steps must be taken.

1. Set the IMPORT_GRDNETCDF_YN to Y to read the native netCDF annual inventory files.

2. Grdmat is used to convert the data from the input data grid (e.g. lat-lon for EDGAR) to the output modeling grid. Set the IOAPI_GRIDNAME_1 to specify the output grid.

3. SMOKE reads each grid cell in the gridded inventory file as a unique source that will have a unique source ID. The lower left corner of inventory grid will be source 1, cell (2,1) will be source 2, cell (3,1) source 3, etc

4. For each gridded inventory file, a Geographical Code Mask Input file (GRIDMASK) is provided by UNC for EDGAR inventory files that assigns a GEOGRID_LEVEL2 value to each grid cell.

5. The ARINV file is used to read in a list of gridded inventory files and to specify the SCC associated with each gridded inventory file

6. Set the NETCDF_POL_UNIT to kg m-2 -s

7. Set the NETCDF_INV_YEAR to the year of the inventory (e.g. 2010)

8. To apply control factors to adjust emissions by GEOCODE_LEVEL and SCC use the CONTROL packet in the GCNTL file

9. To apply vertical profiles to gridded inventory data using the SMOKE Utility Program Layalloc. The EDGAR inventory includes fixed vertical profiles by sector that define the percentages of emissions

by altitude band. Layalloc will create 3-d output files for the gridded emissions sectors that emit above the surface layer (e.g. energy and industrial sources).

4.4.2.8. Files and settings for all inventories

For importing all inventories, the following steps should be taken to review settings and change them as appropriate for your specific modeling case.

Assigns file settings:

1. Change the Assigns files inventory settings.

   Following the settings in Section 4.4.1, "Change Assigns file to set scenario names, grid names, and other case-specific configuration information" [126], you should have created a new Assigns file for your case and set the `INVID`, `INVEN`, and `INVOP` settings described in Table 4.1, "Variable in Assigns file for naming files and directories" [128]. Since the inventory input directories depend on the `INVID` setting, you should make sure that the value set for `INVID` is consistent with the location of the inventory files. For example, the `ARINV` files should be installed in `$SMKDAT/inventory/$IN-VID/area`. Make sure that `$INVID` label in the paths for the files installed in prior subsections is the same as the `$INVID` label in the Assigns file that is included in the Assigns file you will use.

2. Set the `ABASE`, `MBASE`, `PBASE`, and `EBASE` settings described in Table 4.1, "Variable in Assigns file for naming files and directories" [128]. This is usually done for a new inventory, and can also be done for new scenarios with an existing inventory.

Script settings:

3. Double check that the correct Assigns file (e.g., the one with the correct `$INVID` setting) is used by the run scripts for your case.

4. Set the `RAW_DUP_CHECK` setting to Y if you want **Smkinven** to give an error if any duplicate entries are found in the inventory. See Section 6.13, "**Smkinven**" [260] for more information.

5. Set the `SMKINVEN_FORMULA` setting to compute one or more pollutant values from teh values of other pollutants.

   This is most commonly used to compute coarse particulates (PMC) from PM_{10} and $PM_{2.5}$ using the setting "PMC = PM10 - PM2_5".

6. Set the `WEST_HSPHERE` setting to reset any positive longitude values that are intended for the Western hemisphere to a negative value.

7. Set the `WKDAY_NORMALIZE` setting to account for the desired weekday normalization approach to be used for hourly emissions. See Section 6.13, "**Smkinven**" [260] for more information.

8. Set the `SMK_PROCESS_HAPS` to read the `NHAPEXCLUDE` file.

   This setting is only relevant when importing both criteria and toxic inventories. In this case, if there are VOCs in both inventories and the `INVTABLE` has indicate which toxics are VOCs, **Smkinven** will attempt to integrate VOC emissions for all sources from the criteria and toxics inventories. The `NHAPEXCLUDE` file provides a list of SCCs for which **Smkinven** should not integrate the inventories and should instead drop the toxic VOC emissions to prevent double counting. Toxic emissions of explicit model species are never dropped, however. More information is available in Section 2.9.5, "Combine toxics and criteria inventories" [55].

9. Set the `SMK_ARTOPNT_YN` setting to Y when **Smkinven** should use the `ARTOPNT` file to assign point source locations to area-source emissions.

10. Set the `FILL_ANNUAL` setting to Y to fill annual emissions values with average-day values, when average-day values are available but annual values are not.

    See Section 4.4.2.9, "Using average-day and annual data in the same run" [141] for more information on why one might want to use this option.

4.4.2.9. Using average-day and annual data in the same run

The ORL and FF10 formats provide fields for both annual and average-day emissions values. SMOKE will import both values when either of these formats are used and store all emissions in the SMOKE intermediate inventory files. To configure the **Temporal** allocation step, you have the option of using the `SMK_AVEDAY_YN` setting to select using either the annual or average-day data when both are available. This option and its typical usage will be explained with the **Temporal** settings.

This section describes what to do when both annual data and average-day data need to be used in the same SMOKE run. This can occur when part of an inventory has been provided using annual inventories only (the average-day values are missing or incorrect) and a different part with average-day inventories only (the annual values are missing). This is only a problem when you want to run all of the sources in a single run; so, if nonroad and stationary area sources are provided with the two different temporal resolutions, these can just be run with separate SMOKE runs. If two parts of the stationary area source inventory are provided with two different temporal resolutions, then this section can help to setup the modeling correctly.

Two approaches are available in SMOKE to resolve this situation. In the first approach, one can set SMOKE to run using the average-day data, and in the second approach, set to use the annual data. These lists describe these two approaches:

Run all sources using "annual" data

1. Set the `FILL_ANNUAL` setting to Y. This will cause the missing annual data to be filled in with the available average-day data by multiplying the number of days in the year, accounting for leap years.

2. Ensure that the monthly temporal profile assignments made by the `ATREF`, `MTREF`, or `PTREF` files assign uniform monthly profiles to any all sources that have the annual numbers filled in with seasonal values.

3. During temporal processing, set `SMK_AVEDAY_YN` to N. This will ensure that both the annual values that were available in the inventory and the annual values that were computed will be used by the **Temporal** program. Since the monthly temporal profiles for the filled-in annual emissions have been set to uniform, the average-day values will be calculated from the filled-in annual values.

Run all sources using average-day data

1. Set the `FILL_ANNUAL` setting to N. By default, **Smkinven** computes an average-day value when none is provided by dividing the annual value by the number of days in the year, accounting for leap years.

2. During temporal processing, set `SMK_AVEDAY_YN` to Y. This will ensure that both the average-day values that were available in the inventory and the average-day values that were computed will be used by the **Temporal** program.

3. Note that no monthly profiles are applied to average-day emissions, so if monthly variation is needed, it will need to be calculated outside of SMOKE. In this case, it might be easier to run all sources as annual, as described above.

4.4.3. Select which pollutants in the inventory are kept

Users have the option of deciding at run time which inventory pollutants will be kept from an inventory provided in ORL format. The INVTABLE file lists all possible criteria and toxic pollutants. While the example files are fairly comprehensive, it is possible users may need to add pollutants to this file using the INVTABLE file format description. Additionally, users can change which pollutants are kept for further processing in SMOKE. This is done by setting the Keep column of the file to Y for all pollutants that SMOKE should keep, all pollutants in these files must have the Keep column set to Y.

Both the run time and intermediate file sizes can be reduced by keeping only the pollutants that you need to include for modeling purposes. In the bulleted list below, we provide several examples of why a user might want to change which pollutants are kept.

• You want to model additional REMSAD toxics not set up in the example configuration (e.g., atrazine, dioxin).

• You want to import ROG emissions instead of VOC emissions from a criteria inventory.

• You want to import only mercury for CMAQ modeling and drop all other toxic pollutants.

• You want to add a tracer to the inventory and need SMOKE to output the tracer emissions to the model-ready file.

4.4.4. Set fallback stack paramaters

The **Smkinven** program allows you to fill in stack parameters for sources that do not have any. The details of how this is done are described in Section 2.9.9, "Fill in and check point-source stack parameters" [58]. To change the defaults, the PSTK input file to **Smkinven** must be edited to use different stack parameters and/or include different state/county and SCC codes.

To discover what if any default stack parameters are being used for your SMOKE processing, review the **Smkinven** log file, which will contain warnings about all stack parameters that were filled in and changed from the inventory values.

4.4.5. Changing output and intermediate directory names

As described in Chapter 3, *SMOKE Directory Structure* [97] and also Table 4.1, "Variable in Assigns file for naming files and directories" [128], the output and intermediate directory names are based on user-defined environment variables. The following list provides a checklist for all of the possible changes that you might make and where to find more information on how changes would affect your output and intermediate directories. Make sure that the scripts that you are using to run SMOKE use the same Assigns file in which you are making the changes described in this list.

1. Set the INVEN setting in the Assigns file to the label to use in the output inventory path ($SMKDAT/inventory/$INVEN).

2. Decide whether you want to use the same setting or different settings for ABASE, MBASE, PBASE, and EBASE in the Assigns file.

If these are the same, the directory structure will be the "basic" configuration, and if these are different, the structure will be the "advanced" configuration; both configurations are described in Chapter 3, *SMOKE Directory Structure* [97].

   a.   Set the `ABASE` setting to change the paths for stationary area and nonroad mobile source intermediate and output files.

   b.   Set the `MBASE` setting to change the paths for on-road mobile source intermediate and output files.

   c.   Set the `PBASE` setting to change the paths for point source intermediate and output files.

   d.   Set the `EBASE` setting to change the path for the final merged model-ready emissions from all source categories. This will also set the output path for QA reports of anthropogenic sources. See Chapter 3, *SMOKE Directory Structure* [97] for specifics.

3.   Set the `BBASE` setting in the Assigns file to select using BELD3 or BELD4 data. See Section 4.4.15, "Use BEIS3" [166].

4.   Set the `SPC` setting in the Assigns file to change the output paths for all source categories.

   As described in Chapter 3, *SMOKE Directory Structure* [97], the `OUTPUT` directory is based in part on this `SPC` setting. This setting must be coordinated with the selection of the chemical speciation mechanism (see Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150]).

5.   For future year modeling, the output paths will be changed automatically.

   As described in Chapter 3, *SMOKE Directory Structure* [97], the `FYEAR`, `CNTLCASE`, `SMK_FU-TURE_YN`, and `SMK_CONTROL_YN` script settings can have an impact on the output directory paths. When either `SMK_FUTURE_YN` or `SMK_CONTROL_YN` are set to Y, the output path will be changed as described in Chapter 3, *SMOKE Directory Structure* [97] for the growth and control case configuration.

4.4.6. Change which programs are run

The example SMOKE scripts each have several `RUN_<Program>` settings, where the program names fill in the `<Program>` placeholder. For example, the `RUN_<Program>` setting that controls the **Smkinven** program is `RUN_SMKINVEN`. In the scripts, each `RUN_<Program>` setting can be set to either Y for "yes" and N for "no". When a `RUN_<Program>` setting is set to Y, the script will try to run the program associated with the `RUN_<Program>` setting. For example, when `RUN_SMKINVEN` is set to Y in the script, running the script will cause the **Smkinven** program to run.

Table 4.3, "List of `RUN_<Program>` settings and situations to set to Y" [144] is the full list of `RUN_<Program>` settings available and the situations in which each needs to be run.

Table 4.3. List of `RUN_<Program>` settings and situations to set to Y

Setting	Reasons to set to Y to run program
RUN_SMKINVEN	Import raw inventory data and create SMOKE intermediate inventory file for area, nonroad mobile, on-road mobile, or point sources
RUN_SPCMAT	Compute speciation factors and create speciation matrix for area, nonroad mobile, on-road mobile, point sources.
RUN_GRDMAT	Compute gridding factors and create gridding matrix for area, nonroad mobile, on-road mobile, or point sources.
RUN_CNTLMAT	Compute growth factors and create growth matrix for area, nonroad mobile, on-road mobile, or point sources. Compute control factors and create multiplicative control and/or reactivity control matrices for area, nonroad mobile, on-road mobile, or point sources.
RUN_GRWINVEN	Calculate grown and/or controlled inventory and create SMOKE intermediate inventory files for area, nonroad mobile, on-road mobile, or point sources. Required when using SMOKE to create future-year inventories.
RUN_TEMPORAL	Calculate hourly emissions and create intermediate hourly emissions file for area, nonroad mobile, on-road mobile, or point sources.
RUN_ELEVPOINT	Select elevated and PinG point sources.
RUN_LAYPOINT	Calculate plume rise for CMAQ or MAQSIP for any point sources. Calculate plume rise for UAM, REMSAD, or CAM_X for explicit plume-rise sources (i.e., those sources with precomputed plume rise).
RUN_MOVESMRG	Create 3-d model-ready emissions by merging MOVES RPD, RPP and RPV Look-up tables with Spcmat and Grdmat for mobile sources.
RUN_SMKMERGE	Create 3-d model-ready emissions for point sources or all source categories together for CMAQ or MAQSIP. Create 2-d gridded, hourly, speciated emissions for area sources, nonroad mobile sources, on-road mobile sources, point sources, and/or all sources combined in I/O API format. Convert units of biogenic outputs from **Tmpbio** from moles/hr to moles/s for CMAQ. Create ASCII elevated point source file for UAM, REMSAD, or CAM_X.
RUN_SMK2EMIS	Convert 2-d merged I/O API file with all low-level emissions from **Smkmerge** to binary input format for UAM, REMSAD, or CAM_X.
RUN_SMKREPORT	Create default and/or custom reports of processed emissions.
RUN_NORMBEIS3	Import gridded BELD3 landuse for use in creating BEIS3 emissions.
RUN_TMPBEIS3	Calculate BEIS3 emissions in gridded, hourly, speciated I/O API format.

4.4.7. Set up country codes, state codes, or county codes

By default, SMOKE is setup to use the country, state, and county codes based on a U.S. view. The codes assume a 3-tier system of codes as described in Section 2.3.4, "Country, state, and county codes" [13]. The three tiers allow specification of regions as follows:

- One digit for country code

- Two digits for state codes

- Three digits for county codes

All codes provided in the emission inventories must be integers and must match entries in the COSTCY or GEOCODE_LEVEL4 (if USE_EXP_GEOCODE Y) file, which is the master list of country, state, county and tribal codes in any given SMOKE installation. The format for the COSTCY file can be found in Chapter 8, *SMOKE Input Files* [345]. By default, the codes are shared by all scenarios installed in the same SMOKE system (using different COSTCY or GEOCODE_LEVEL[1-4] (if USE_EXP_GEOCODE Y) files requires changing the Assigns file to not share the same COSTCY or GEOCODE_LEVEL[1-4] (if USE_EXP_GEOCODE Y) file among all cases).

The codes in the inventory must match the codes in this file. SMOKE assigns time zones, daylight saving time exemptions, state names, and county names using this file. Therefore, if one of the counties in the inventory does not match a valid country, state, and county code in the COSTCY or GEOCODE_LEVEL[1-4] file (if USE_EXP_GEOCODE Y), it can cause errors in the modeling.

The following list provides the steps that users should take to ensure that the file will work for their modeling case:

1. Check whether the countries with data in the inventory are included in the /COUNTRY/ section of the COSTCY or GEOCODE_LEVEL[1-4] (if USE_EXP_GEOCODE Y) file.

2. Change country codes and names if necessary within the limitation of 10 countries per file. If the country codes change, the country-specific SMOKE temporal profile assignments made by ATREF, MTREF, and PTREF may no longer be valid.

3. Check whether the states with data in the inventory are included in the /STATE/ section of the COSTCY or GEOCODE_LEVEL[1-4] file (if USE_EXP_GEOCODE Y) file.

4. Change or add state codes and names if necessary to ensure all states are listed for all countries.

5. Check whether the counties with data in the inventory are included in the /COUNTY/ section of the COSTCY or GEOCODE_LEVEL[1-4] file (if USE_EXP_GEOCODE Y) file and that all counties have assigned time zones.

   Smkinven will perform this function for you if needed and will list in the **Smkinven** log file all counties that are not included in the COSTCY or GEOCODE_LEVEL[1-4] file (if USE_EXP_GEOCODE Y) file or that do not have time zones assigned. If county codes have changed for merged or separated counties (e.g., Miami/Dade county in Florida).

6. Change or add county codes and names if necessary to ensure all counties are included in the inventory for all states and countries and that time zones have been assigned to each.

4.4.8. Set up list of known SCCs and ORIS IDs

The descriptions of the SCCs and ORIS IDs in SMOKE can be updated easily by modifying or adding to the files that contain these codes. The SCC descriptions are used in **Smkreport** output reports, and the ORIS ID descriptions are used in **Smkinven** reports when using CEM data. Updating the files descriptions or valid codes is simply a matter of editing the ASCII files that contain these codes. The logical file name for the SCC description file is SCCDESC, and that for the ORIS ID description is ORISDESC. Example SCCDESC and ORISDESC files are in the $GE_DAT directory and described in Section 4.2, "Test case descriptions" [117].

4.4.9. Use a new modeling grid or change spatial inputs (anthropogenic and biogenic)

Using a new modeling grid and/or changing the spatial inputs requires a number of changes to the Assigns file, scripts, and input files. In this section, we first describe how to set up for a different modeling grid and then how to install new inputs for that grid or change the spatial inputs for the existing grid.

4.4.9.1. Set up for a different modeling grid

The following steps are needed to set up for a different modeling grid.

1. Insert or find the grid parameters in the `GRIDDESC` file

    The `GRIDDESC` file contains many grid descriptions and grid projections. Each grid description is defined by a 16-character (or less) name. Chapter 8, *SMOKE Input Files* [345] contains the file format. If the grid you want is already in the file, note its name.

    If the grid you want is not in the file, use the format in Chapter 8, *SMOKE Input Files* [345] to insert or modify a grid description for the new grid. The grid cells must align with the cells of the cross-grid meteorology data files that will be used for modeling biogenic emissions and possibly for point and on-road mobile sources.

2. Change the `GRID` variable in the Assigns file to a new grid name

    As described in Section Section 4.4.1, "Change Assigns file to set scenario names, grid names, and other case-specific configuration information" [126], change the `GRID` environment variable to a name that will be used to name output and intermediate files. This name should be coordinated with the selection made in step 3. Since the `GRID` variable is used in naming files, it is helpful to use a name that is easily distinguished from other grid names that you might use, but not so long as to add a lot of characters to the name of each file that includes the `GRID` variable in its name. Usually a name of 4 or 5 characters is sufficient.

    Changing the `GRID` setting is a good reason to create a new Assigns file. To do this, you can copy an existing Assigns file as described in Section 4.4, "How to use SMOKE" [125].

3. Change the `IOAPI_GRIDNAME_1` variable in the Assigns file to the name of the grid entered or found in step 1.

    As noted in step 2, the name selected should have some relationship to the `GRID` environment variable. For example, if the `IOAPI_GRIDNAME_1` setting is "US36_120X90", the `GRID` setting might be "us36".

4. Update the run scripts to use the Assigns file for the new grid.

5. Follow the remaining steps in the next subsection.

4.4.9.2. Installing or changing spatial input files

The following steps are needed to install or change the spatial input files.

1. Update or install the grid description in the `GRIDDESC` file, if not done previously (see step 1 in the previous subsection).

2. Create and install the spatial surrogates file and surrogate description file `SRGDESC`

The Area and Mobile spatial surrogates files located in `SRGPRO_PATH` and the `BGPRO` are one of the most difficult inputs to create for SMOKE. The gridded data in the files must match the definition in the `GRIDDESC` and `SRGDESC` files for the grid selected using the `IOAPI_GRIDNAME_1` variable.

The best approach to creating this file is to obtain it from someone else who has created surrogates for your domain. For example, EPA has developed various spatial surrogates and available at EPA's Emission Modeling Clearinghouse (http://www.epa.gov/ttn/chief/emch/ [http://www.epa.gov/ttn/chief/emch/index.html]). Of course, using a national file has its limitations if your domain happens to be far away from the center of the U.S. since the map projection for the surrogates is more distorted further away from the center of the projection. However, for most users, this distortion will be a small price to pay for having readily available spatial surrogates.

The second best approach to creating this file is to use the MIMS Spatial Allocator (MSA). This tool can create SMOKE-ready surrogates using GIS shape files, which are provided with the tool for multiple surrogates. Please refer to the CMAS web site [http://www.cmascenter.org] for more information.

A third approach to creating this file it to perform the same calculations done by the MSA using a GIS like ArcInfo. The calculations being done by the MSA are provided on the MSA website and could be implemented in a GIS, though this would probably be much too difficult an option for most users.

The surrogate file does not need to have the same outer boundaries as your modeling grid, but the grid cells do need to align with the grid cells of the surrogate file. This means that the projection of the surrogates must be the same as the projection as the modeling grid and the cell sizes must also match. SMOKE will produce an error if there is an inconsistency. If the grid represented by the surrogates file is larger than the modeling grid and the cells align, SMOKE will automatically extract the grid cells for the modeling domain from the larger grid provided by the surrogates.

In addition to the surrogate file, it is important to create a surrogate description file (`SRGDESC`). This is simply a file that lists the surrogate codes and their descriptions. Since the surrogate file itself does not currently have descriptive text that indicate what the surrogate codes mean, the surrogate description file is the only way users have of knowing to what data the codes in the surrogate file refer.

The same surrogate file can be used for Area and Mobile. This is accomplished by using the same physical file name in the `SRGDESC`. Please see Chapter 8, *SMOKE Input Files* [345] for more information on these files. The `BGPRO` file must include the "county area" spatial surrogate, and is used for estimating county-total emissions from gridded emissions.

The Area and Mobile spatial surrogate file(s) should be installed as a sub-directory in the `$GE_DAT` directory. This sub-directory is assigned to `SRGPRO_PATH`. The surrogate description file should be stored in `$GE_DAT` as well. It is useful to include the name of the `GRID` in the name of the sub-directory `SRGPRO_PATH` for you may in the future be modeling different grids and therefore needing different surrogate files. The Biogenic surrogate file `BGPRO` should be located in `$GE_DAT`.

3. Create and install the spatial cross-reference file

An example spatial cross-reference file (`AGREF` and `MGREF`) is provided with SMOKE that can be used for most inventories as long as the spatial surrogate codes in the new surrogate file match the codes in the example surrogate file, which are provided in the `SRGDESC` file described in Chapter 3, *SMOKE Directory Structure* [97]. If the codes match then the only concern will be whether the surrogate assignments are acceptable. By default, the same file is used for both the `AGREF` and `MGREF` files, which is accomplished by using the same physical file name in the Assigns file for both logical file names. The files can be the same file or two different files, at the discretion of the user - usually

one file is more convenient unless different default settings (SCC=0) are needed, or unless the data all already available as two separate files.

When **Smkinven** imports an area, non-road mobile, or on-road mobile inventory, a list of SCCs is created (logical file name `ASCC`, `NSCC`, or `MSCC`). After importing the data but before running spatial allocation, it is wise to examine the list of SCCs included in this file and compare it with the list in the example spatial cross-reference files. This comparison can be accomplished manually or using the **Smkreport** program with a report that includes the surrogate codes assigned to each SCC (see Chapter 7, *SMOKE Quality Assurance* [305]). The **Smkreport** results will list all of the surrogate as-signments by SCC and the emissions associated with each, with which users may evaluate if any SCCs were included in the inventory but not in the example SMOKE file. The example spatial cross-reference files (`AGREF` and `MGREF`) can be updated with any new assignments or to include SCCs that are in the user's inventory but not in the example files.

If a new surrogates file uses surrogate codes that are different from the example codes, then new spatial surrogate cross-reference files need to be created. These can be created by copying and editing the existing files to replace the example surrogate codes with the new codes. For example, the current code for population is 100. If the new code for population is 400, then all of the "100" surrogate codes in the file would need to be replaced with "400"s. These replacements must be made carefully because the example file also uses "400" for the rural land area surrogate, so if global search and replace is performed, one must be careful to change what one intends to change. It is much easier to simply use the codes that are already in use in any new surrogates file, or modify the surrogate file to use those codes.

The spatial cross-reference file(s) should be installed in the `$GE_DAT` directory, since the file is likely to be shared by many SMOKE runs and source categories. It is useful to name the file(s) to include the `GRID` environment variable for easier identification as association with the surrogate file(s) with which it works.

4. Create and install the biogenic gridded land use data

You must create a gridded biogenic land use data (`BELD4` for BEIS3.60 or `BELD3_A`, `BELD3_B`, and `BELD3_TOT` for BEIS3.14) to use for your grid. The MSA described in step 2 is able to create these files using the BELD3 or BELD4 database. Please refer to step 2 for the web address of this tool. The outputs of the MSA can be used for BEIS3 modeling directory.

The gridded land use data in the file must match the definition in the `GRIDDESC` file for the grid se-lected using the `IOAPI_GRIDNAME_1` variable. As with the spatial surrogates, the grid can be larger than the modeling grid, but it the cells must overlay exactly with the grid cells of the actual modeling grid, which includes the grid projection and cell sizes.

The gridded biogenic land use file(s) should be installed in the `$BGDAT` directory for BEIS3 modeling.

5. Create and install the meteorology files.

Meteorology files are a necessary input to the air quality model; therefore, the same meteorology files that will be used for the air quality modeling can be used for input to SMOKE. The MCIP program (a CMAQ preprocessor) can be used to convert the files from MM5 output format to the I/O API input format for SMOKE and CMAQ. Other user-created programs can be used as well as long as the variable names and file arrangements are consistent with what SMOKE expects. Chapter 8, *SMOKE Input Files* [345] explains the details on the meteorology files that SMOKE expects and the variables that must be included in these files.

The meteorology files must be consistent with the grid selected. Like the surrogate file, the meteorology grid can be larger than the SMOKE modeling grid, but the grid cells must align. This includes the

cell sizes being the same and the projection being the same. If this is the case, the SMOKE programs that use the meteorology files (**Met4moves**, **Temporal**, **Laypoint**, **Tmpbio**, and **Tmpbeis3**) will extract the meteorology data from the region of the files that intersects with the modeling grid.

6. Optionally create the BIOSEASON file

   If you are modeling biogenic emissions with SMOKE for an episode and grid that covers both winter and summer seasons, then the best modeling approach is to create a BIOSEASON file from the meteorology data for input to **Tmpbio** or **Tmpbeis3**. This file indicates which grid cells and days are during "winter" and which are during "summer", to allow **Tmpbio** or **Tmpbeis3** to apply the winter or summer emission factor for the correct cells and days. Summer is defined for biogenic modeling in the Northern Hemisphere as the period after the last freeze and before the first freeze in each grid cell. This file may not apply to some episodes or regions which do not experience freezing temperatures, and in those cases, users may run **Tmpbio** or **Tmpbeis3** without this file.

7. Set the fallback surrogate in the run scripts for area/non-point, nonroad mobile, and on-road mobile sources.

   All scripts for source categories that use spatial surrogates have a **Grdmat** setting SMK_DEFAULT_SR-GID that is used to set a fallback surrogate code. The fallback surrogate code is explained in Chapter 6, *SMOKE Core Programs* [211] with the **Grdmat** program. It prevents emissions from being zeroed out when there is no surrogate of the assigned type for a source. The fallback surrogate should therefore be a surrogate that is never zero for a county, such as population. Only one fallback surrogate can be set for all sources in a given run. The SMK_DEFAULT_SRGID setting should be reset if the surrogate codes change (e.g., the default setting of 8 is not population) or if you want to change the fallback surrogate to a different one.

8. Update Assigns file to use all new files.

   Update the Assigns file that you are using for this case to include all new files that you have created in this step:

   - BGPRO

   - AGREF, MGREF

   - SRGDESC

   - SRGPRO_PATH

   - BGUSE

   - BELD3_A, BELD3_B, BELD3_TOT

   - BIOSEASON

9. Check that the inventory files cover the region of the grid.

   When using a new grid, it is important to use an inventory that covers the states and counties included in your domain. There are two ways of checking that this is the case. First, obtain a map of your modeling domain prior to creating your inventory and note the states that are included wholly or in part in the domain. Before modeling, create, adapt, or amend your inventory to ensure that the inventories from each of those states are included.

   The second way that you can check that you are using data for all regions of the domain is to run the data through SMOKE and look at the resulting gridded emissions in PAVE. Using PAVE, reset the

scale of the plot to a very small maximum value (e.g., 0.001). All areas of the domain with emissions should show up red. If any areas show up gray, then there are missing values that could be because the inventory is not complete.

4.4.10. Use a different speciation mechanism or change speciation inputs

A common change to the SMOKE configuration is to select a speciation approach for an air quality model and chemical mechanism. In Chapter 4, *Using SMOKE Scripts* [117], we noted that many speciation files are available for download from EPA. In this section, we describe how to use the `SPC` value in the Assigns file to select these.

4.4.10.1. Choosing from default mechanisms

SMOKE provides several chemical mechanisms that one can select for modeling. In the default SMOKE configuration, only a few small changes are necessary to the Assigns file and script to use a different chemical mechanism.

1. Set the `SPC` setting in the Assigns file.

    The `SPC` setting in the Assigns file controls which chemical mechanism is used by the scripts and SMOKE programs. Depending on what source categories you are modeling, the air quality model that you are using, and the chemical mechanism for that air quality model, choose the correct setting from `SPC` from Table 4.4, "Valid `SPC` settings" [151]. Use the descriptions below the table to decide if the `SPC` setting will work for your case. See the default data files described in Chapter 4, *Using SMOKE Scripts* [117] for the actual file names that will be used for each source category. Where different files are needed for different source categories, the Assigns file will automatically detect this and set the `GSPRO` and `GSREF` files accordingly.

Table 4.4. Valid `SPC` settings

SPC Setting (chemical mechanism)	Air Quality Models	Inventories	BEIS3	Non-point/area	Nonroad mobile	On-road mobile	Point
cmaq.cb5p25 (CB5 with $PM_{2.5}$ speciation)	CMAQ MAQSIP* UAM* CAM_X*	criteria	X	X	X	X	X
cmaq.cb5p25_wtox (CB5 with $PM_{2.5}$ speciation, but inputs can include VOC toxics)	CMAQ MAQSIP* UAM* CAM_X*	criteria toxics	X	X	X	X	X
cmaq.cb5tx1p25 (CB5 toxics version 1, with $PM_{2.5}$ speciation)	CMAQ	criteria toxics	X	X	X	X	X
cmaq.radm2p25 (RADM2 with $PM_{2.5}$ speciation)	CMAQ	criteria	X	X	X	X	X
cmaq.saprc99pm (SAPRC99 with $PM_{2.5}$ speciation)	CMAQ	criteria	X	X	X	X	X
maqsip.radm2pm	MAQSIP	criteria	X	X	X	X	X
remsad.cb4mpm	REMSAD v6	criteria	X	X	X	X	X
remsad7.cb4mpm	REMSAD v7	criteria toxics	X	X	X	X	X

*Does not work for SO_2, NH_3, $PM_{2.5}$, and PM_{10} for this model and/or mechanism. These pollutants must not be included in the inventory when using this speciation setting for these models.

- cmaq.cb4p25:

    - Use for CMAQ, MAQSIP, UAM, and CAM_X for Carbon Bond 4 only (when Keep = N for all PM pollutants in `INVTABLE` file or inventory does not include PM pollutants)

    - Use for CMAQ for Carbon Bond 4 VOC speciation and $PM_{2.5}$ speciation (no mercury)

    - Use for BEIS3 modeling for all models for CB5

- cmaq.cb4p25_wtox:

    - Use for CMAQ for CB5 and PM, and for MAQSIP, UAM, and CAM_X for CB5 only

    - Modeling with both criteria and toxics inventories for the standard CB5 chemical mechanism. The on-road mobile `GSPRO` file is different from that of the other source categories.

    - Use for BEIS3 modeling for all models for CB5.

- cmaq.cb4tx1p25:

- Use for CMAQ modeling only with both criteria and toxics inventories for the "toxics-CB5" chemical mechanism. This works for all source categories, including BEIS3 modeling. The `GSPRO` and `GSREF` files are different for on-road mobile sources, and the `GSREF` file is different for the point sources.

- cmaq.radm2p25:

  - Use for CMAQ with RADM2 chemical mechanism, with or without PM emissions. This works for all source categories, including BEIS3 modeling.

- cmaq.saprc99pm:

  - Use for CMAQ with SAPRC 99 chemical mechanism. This works for all source categories, including BEIS3 modeling. The `GSPRO` file is different for on-road mobile sources.

- maqsip.radm2pm:

  - Use for MAQSIP with RADM2 chemical mechanism, with or without PM emissions. This works for all source categories, including BEIS3 modeling.

- remsad.cb4mpm:

  - Use for REMSAD version 6 with micro-CB5 and PM emissions. This works for all source categories, including BEIS3 modeling.

- remsad7.cb4mpm:

  - Use for REMSAD version 7 for micro-CB5 and speciated $PM_{2.5}$ emissions. Use also for mercury and cadmium modeling. This works for all source categories, including BEIS3 modeling.

2. Determine whether VOC-to-TOG or ROG-to-TOG is needed and set script accordingly.

   If your inventory contains the pollutant VOC or ROG and you are modeling for CMAQ, MAQSIP, UAM, or CAM_X, then you will need to use SMOKE's `GSCNV` file to convert the VOC or ROG mass to TOG mass.

   The SMOKE **Spcmat** program will use the `GSCNV` file if the `POLLUTANT_CONVERSION` setting in the script is Y. This is the default configuration for the CMAQ run scripts.

   If you are using one of the REMSAD models or if your inventory contains TOG, then the `POLLUTANT_CONVERSION` setting in the run script should be set to N.

3. Ensure that the inventory pollutants that need speciation match the inventory pollutants in the speciation profiles and cross-reference files.

   All of the pollutants in your inventory that you wish to include in the outputs must have valid profiles in the `GSPRO` and `GSREF` files. If the pollutant is left out of the `GSREF` file, the pollutant will be dropped from modeling (in some cases, this is desirable, but a warning will be generated by **Spcmat**). For each pollutant in your inventory check that:

   a. The pollutant has one or more entries in the `GSREF` file. In some cases, you may have to add entries. For example, not all `GSREF` files have "ROG" entries, but these can easily be added simply by copying the VOC assignments and changing the VOC label to ROG.

b. For CMAQ, UAM, and CAM$_X$ and the pollutant is VOC or ROG, check that the TOG profiles are in the GSPRO file.

c. For REMSAD and the pollutant is VOC, check that the VOC profiles are in the GSPRO file.

d. For all models and all other pollutants, check that the profiles are included in the GSPRO file.

4. Ensure that the correct Assigns file is used in your run scripts.

4.4.10.2. Setup of new mechanism or changing an existing mechanism

1. Create and install the GSPRO file.

    If only a few changes will be made to a GSPRO file (like adding some more toxic species for REM-SAD), you can simply copy an existing GSPRO file to a new file. If more extensive changes are needed to create a new mechanism, currently, there is no utility for creating a GSPRO file in SMOKE. For now, users must use the file format in Chapter 8, *SMOKE Input Files* [345] and a high degree of expertise to develop new GSPRO files.

    Install the GSPRO file in the $GE_DAT directory. For the default GSPRO file for that chemical mechanism (i.e., the one that applies to the most source categories), name the file gspro.$SPC.txt. If your speciation profile is specific to a certain source category, name the GSPRO file as follows:

    • area/non-road mobile: gspro.$SPC.a.txt

    • on-road mobile: gspro.$SPC.m.txt

    • point: gspro.$SPC.p.txt

2. Create and install the GSREF file.

    You may be able to simply copy a GSREF file to a new name, depending on whether the profile codes are different for your new mechanism. In many cases, the same profile codes are used (e.g., from the SPECIATE database), but the profiles themselves are all that have changed.

    If you need to create a new GSREF from scratch, you will need to ensure that all of the SCCs and all of the pollutants in the inventory have profile assignments. Any pollutant in the inventory that is not included in the GSREF file will be dropped from modeling. Any SCC that is not included will be assigned the default profile, which is the profile assigned to SCC = 0.

    Name the GSREF file using a similar approach as for the GSPRO file, but use the "gsref" prefix instead of the "gspro" prefix. Therefore, the default file for that chemical mechanism (the one that applies to the most source categories) should be named gsref.$SPC.txt. Any source-category-specific files should use the names as follows:

    • area/non-road mobile: gsref.$SPC.a.txt

    • on-road mobile: gsref.$SPC.m.txt

    • point: gsref.$SPC.p.txt

3. Create the pollutant-to-pollutant conversion file, if necessary.

    If the speciation profiles change or new profiles are added for VOC speciation, a GSCNV file must be created accordingly. This can be accomplished by copying and editing the example GSCNV file or

creating a new `GSCNV` file from scratch. This requires a high degree of expertise and must be closely coordinated with creating the `GSPRO` file.

4. Edit the `INVTABLE` file.

Particularly for toxics modeling, changes to the `GSPRO` file must be coordinated with changes to the `INVTABLE` file in the following ways:

- If the definition of NONHAPVOC changes, the `INVTABLE` definition of NONHAPVOC must change as well.

- If there are new explicit model species, these must be identified in the `INVTABLE` file.

- If there are new model species that are not part of the chemical mechanism, these must be identified in the `INVTABLE` file.

- If there are new pollutants, these must be included in the `INVTABLE` file (as well as the raw inventories, of course).

5. Coordinate with MOVES inputs.

When processing using on-road mobile emissions and SMOKE running MOVES, users need to create the `MEPROC` file that contains a list of MOVES pollutants by emissions processes created by MOVES. This SMOKE input need to be coordinated so that the pollutants expected by the chemical mechanism are all created by SMOKE/MOVES. This requires that the `MEPROC` file have all pollutant/process combinations for the required pollutants and that the MOVES inputs are configured to generate emissions for those pollutants. The `MEPROC` file is documented in Section 8.9.2.1.3, "`MEPROC`: List of mobile emission processes and associated activities and pollutants" [426]. The MOVES inputs are documented in the MOVES manual, and the limitations placed on these files by SMOKE are documented in Section 4.4.12, "Use MOVES for on-road mobile sources" [159].

6. Edit the Assigns file and scripts

Make sure to change the `SPC` variable in your Assigns file to the one for your new chemical mechanism. Also make sure that your run scripts use the Assigns file with the correct `SPC` setting.

4.4.10.3. Changing speciation assignments

Speciation assignments can be changed so that different speciation profiles are used for certain SCCs and pollutants or to add SCCs and pollutants. You will need to ensure that all of the SCCs and all of the pollutants in the inventory have profile assignments. Any pollutant in the inventory that is not included in the `GSREF` file will be dropped from modeling. Any SCC that is not included will be assigned the default profile, which is the profile assigned to SCC = 0.

1. Adding entries to an existing cross-reference file.

Edit an existing cross-reference file or copy it to a new file for editing. Insert the new speciation assignments based on the format of the file, described in Chapter 8, *SMOKE Input Files* [345].

2. Create new speciation cross-reference file(s) for your case.

Copy the speciation cross-reference (`GSREF`) file for the chemical mechanism that you are using (or a chemical mechanism that is most similar to the one you have created). Edit the speciation profile assignments using the file format, described in Chapter 8, *SMOKE Input Files* [345]. The naming convention for the new file must be consistent with the convention described in the previous subsection.

3. Change the pollutant-to-pollutant conversion file.

    If different SCCs get different profiles, then the `GSCNV` file must be changed as well. The VOC-to-TOG factor in the `GSCNV` depends on the profile that has been assigned to an SCC. So, the factor must be consistent with the profile assigned. Calculating the VOC-to-TOG factor requires some expertise, and this is outside the scope of this manual. However, if you change an SCC to use an existing profile, you can determine the correct VOC-to-TOG factor by first finding another SCC that uses that profile and then setting the VOC-to-TOG factor to be the same as the one for that other SCC.

4. If any new files are created, ensure that the Assigns file uses all new files.

    Make sure that the `SPC` setting in the Assigns is correct and that the Assigns file will set the `GSPRO`, `GSREF`, and `GSCNV` files so that any new files will be used by the programs.

4.4.10.4. Changing the definition of NONHAPVOC or NONHAPTOG

The definition of NONHAPVOC and NONHAPTOG simply must be consistent between the `GSPRO` file and the `INVTABLE` file. This change applies only to chemical mechanisms that use VOC model species and can therefore benefit from the integration of toxics and criteria VOC. This includes all models that SMOKE supports except REMSAD.

1. Edit the `INVTABLE` file.

    You can add pollutants and change their NONHAPVOC status using the `INVTABLE` file format provided in Chapter 8, *SMOKE Input Files* [345].

2. Create new speciation profiles.

    The speciation profiles must use the same definition of NONHAPVOC or NONHAPTOG. The header of the file must be updated to reflect the new definitions. The profiles for NONHAPVOC and NONHAPTOG must also be updated to be consistent. Finally, all other TOG profiles must be updated to be consistent with the definition.

3. Coordinate with MOVES inputs.

    When processing using on-road mobile emissions and SMOKE running MOVES, users need to create the `MEPROC` file that contains a list of MOVES pollutants by emissions processes created by MOVES. This SMOKE input need to be coordinated so that the pollutants expected by the chemical mechanism are all created by SMOKE/MOVES. This requires that the `MEPROC` file have all pollutant/process combinations for the required pollutants and that the MOVES inputs are configured to generate emissions for those pollutants. The `MEPROC` file is documented in Section 8.9.2.1.3, "`MEPROC`: List of mobile emission processes and associated activities and pollutants" [426]. The MOVES inputs are documented in the MOVES manual, and the limitations placed on these files by SMOKE are documented in Section 4.4.12, "Use MOVES for on-road mobile sources" [159].

4.4.11. Use projection

4.4.11.1. Setup for using growth only

The SMOKE growth capability is needed when the inventory year and the modeling year are different. For example, if your base-year inventory is for 2002 and you must evaluate a control strategy in 2010, then you will need to use the SMOKE growth capability.

1. Create and install growth packet files

The /PROJECTION/ packet handles the growth from the base year to a future year. This packet is part of the GCNTL file described in Chapter 8, *SMOKE Input Files* [345]. A packet for each source category that requires growth should be created and saved in a file with the correct name, as described next. Users must ensure that all of the SCCs in the inventory that require growth are included in this packet. Any source that does not match an entry in this file will have a factor of 1 assigned to indicate no change from the base to the future year.

In the default configuration of SMOKE, a GCNTL file must be created that includes the /PROJECTION/ packet to use SMOKE growth. This file must follow the naming convention gcntl.$YEAR_$FYEAR.txt, where YEAR is set in the Assigns file and FYEAR is set in the script as per Section 4.2.4, "Script settings" [121].

The file should be installed in the inventory input directory for the source category to which it applies:

- Non-point (stationary area) files: $ARDAT directory

- Nonroad mobile files: $NRDAT directory

- On-road mobile files: $MBDAT directory

- Point files: $PTDAT directory

Please note that these instructions apply to on-road mobile sources in the following ways:

- Growth of on-road mobile VMT

- Growth of on-road mobile precomputed emissions

Creating on-road mobile emission factors is accomplished by changing the FYEAR setting and using the SMK_FUTURE_YN set to Y. This is described also in Section 4.4.12, "Use MOVES for on-road mobile sources" [159] on running SMOKE with MOVES.

2. Create scripts with correct settings for your particular needs

    To create the scripts with the correct settings, the best starting place is one of the SMOKE example future-year scripts, introduced in Section 4.2.3, "Example script files and their purposes" [120]. These scripts are already setup to run for growth only. To help you understand what the scripts are doing, we have described in the list below the key elements of the script:

    - Uses the base-year Assigns file for the case we are running. The script settings (such as FYEAR and SMK_FUTURE_YN) will adjust what the Assigns file does.

    - FYEAR defined and SMK_FUTURE_YN not defined initially (default is N).

    - Run **Cntlmat** to create growth matrices only (these scripts do not create control matrices, although that is not normally what users will do in real cases). This program reads the GCNTL file and creates a growth matrix.

    - Run **Grwinven** to grow base year inventory to future year (no controls).

    - Reset SMK_FUTURE_YN to Y (note: placeholder for resetting SMK_CONTROL_YN is included, but is not set to Y in example script).

    - Source the Assigns file, which will change output directories and file names now that SMK_FU-TURE_YN is set to Y. Note that RUN_PART1 is temporarily reset to N in the example script to prevent Assigns file from deleting SMOKE intermediate files.

- Run **Temporal** and **Smkmerge**. The **Temporal** program will use the grown emissions to create hourly grown emissions. **Smkmerge** will use these in addition to the base-year gridding matrix and speciation matrix to create future-year model-ready emissions. The base year matrices can be used because they are identical to the future-year ones when SMOKE is used to apply growth.

4.4.11.2. Setup for using controls only

Using controls only in the SMOKE scripts is needed when you import a grown inventory that does not include any or all of the controls that you wish to apply to the inventory. This is what we have assumed in these instructions. We assume that in this case, the growth of the inventory was performed outside of SMOKE, and SMOKE has read in an inventory for the year defined by the FYEAR script setting. This inventory must be imported in a different script that is similar to the base-year script, but follows the instructions for importing future-year inventories in Section 4.4.2.6, "Future-year inventories" [137].

1. Determine which control packets you need

   Please refer to the **Cntlmat** section of Chapter 6, *SMOKE Core Programs* [211] for a description of what each of the control packets can do and how they interact.

   The available packets are:

   - /MACT/ packet

   - /CONTROL/ packet

   - /EMS_CONTROL/ packet

   - /CTG/ packet

   - /ALLOWABLE/ packet

   - /REACTIVITY/ packet

2. Create and install control packets file

   The control packets just listed handle the application of controls for a certain future year. These controls can represent the existing controls that must be applied by emission sources so that they will be in compliance with the law. The controls can also represent additional controls that you want to evaluate to see their impact on modeled air quality. Chapter 8, *SMOKE Input Files* [345] explains these packets and their formats.

   You should create a separate control file for each source category that contains all of the control packets that apply to that source category. Any inventory source and pollutant that does not receive a control is assigned a factor of 1 to indicate no change between the base case and the control case.

   In the default configuration of SMOKE, a GCNTL file must be created that includes the only control packets. This file must follow the naming convention gcntl.$CNTLCASE.txt, where CNTLCASE is set in the script as per Section Section 4.2.4, "Script settings" [121].

3. Create scripts with correct settings for your particular needs

   To create the scripts with the correct settings, the best starting place is one of the SMOKE example future-year scripts, introduced in Section 4.2.3, "Example script files and their purposes" [120]. The script must be changed, however, in the following ways to implement control-only. There are two ways to run controls in SMOKE. The first is to apply multiplicative controls using **Grwinven** and

reactivity controls (if needed) with **Smkmerge**. The second is the apply both multiplicative and reactivity controls using **Smkmerge**. The following settings must be used for each case:

- Apply multiplicative controls with **Grwinven** and reactivity control (if any) with **Smkmerge**

    - Uses the base-year Assigns file for the case we are running. The script settings (such as FYEAR, CNTLCASE, and SMK_CONTROL_YN) will adjust what the Assigns file does.

    - Set FYEAR = the imported future-year inventory

    - Set SMK_FUTURE_YN = Y on the line after FYEAR. Remove this setting from later in the script (around line 120).

    - Set RUN_CNTLMAT = Y to run **Cntlmat** to create control matrices only

    - Set RUN_GRWINVEN = Y to run **Grwinven** to import the control matrices

    - Set RUN_TEMORAL = Y to run **Temporal** on controlled inventory

    - Set RUN_SMKMERGE = Y to run **Smkmerge** on controlled inventory

    - Set options for **Cntlmat**, **Grwinven**, **Temporal**, and **Smkmerge** (see Chapter 6, *SMOKE Core Programs* [211]).

    - Set SMK_NUM_CTLMAT to 1

        For area or nonroad mobile, set ACMAT01 to $ACMAT

        For on-road mobile, set MCMAT01 to $MCMAT

        For point, set PCMAT01 to $PCMAT

        These settings assume that you are using only one multiplicative control matrix. **Grwinven** can actually import multiple control matrices from different cases. To use more than one control matrix, set SMK_NUM_CTLMAT to the number of control matrices and set ACMAT02, ACMAT03, etc to the names of the additional control matrices.

    - Set CNTLCASE to the control strategy name that you want for the control case.

    - At line 120 of file, reset SMK_CONTROL_YN to Y

    - To run **Smkmerge** for reactivity controls, add the MRG_CTLMAT_REAC option to the **Smkmerge** options and set to "A", "M", "P" or some combination of those letters to indicate whether you want it to use the reactivity matrix for area (including nonroad), on-road mobile, or point sources.

- Apply both multiplicative and reactivity controls with **Smkmerge**

    - Uses the base-year Assigns file for the case we are running. The script settings (such as FYEAR, CNTLCASE, and SMK_CONTROL_YN) will adjust what the Assigns file does.

    - Set FYEAR = the imported future-year inventory

    - Set SMK_FUTURE_YN = Y on the line after FYEAR. Remove this setting from later in the script (around line 120).

    - Set RUN_CNTLMAT = Y to run **Cntlmat** to create control matrices only

- Set RUN_GRWINVEN = N to run **Grwinven** to import the control matrices

- Set RUN_TEMORAL = N

- Set RUN_SMKMERGE = Y to run **Smkmerge** on controlled inventory

- Set options for **Cntlmat** and **Smkmerge** (see Chapter 6, *SMOKE Core Programs* [211]).

- Set CNTLCASE to the control strategy name that you want for the control case.

- At line 120 of file, reset SMK_CONTROL_YN to N

- To run **Smkmerge** for multiplicative controls, add the MRG_CTLMAT_MULT setting to **Smkmerge** options and set to "A", "M", "P" or some combination of those letters to indicate whether you want it to use the multiplicative control matrix for area (including nonroad), on-road mobile, or point sources.

- To run **Smkmerge** for reactivity controls, add the MRG_CTLMAT_REAC setting to the **Smkmerge** options and set to "A", "M", "P" or some combination of those letters to indicate whether you want it to use the reactivity control matrix for area (including nonroad), on-road mobile, or point sources.

4.4.11.3. Setup for both growth and controls

Follow the instructions that apply for both of the previous two subsections. The differences between this case and the previous two cases are as follows:

- Name the GCNTL input file differently. It should be named: gcntl.$YEAR_$FYEAR_$CNTL-CASE.txt

- The **Grwinven** program must be used at least to grow the inventory, so you must

  - Set RUN_GRWINVEN to Y.

  - Set SMK_NUM_CTLMAT to at least 1. It should be set to the total number of growth and control matrices to be applied by the **Grwinven** program.

  - The ACMAT01, ACMAT02, etc file names should be set to be consistent with the matrices being applied and the source categories (e.g., ACMAT01 should be $APMAT or $ACMAT, but not $PCMAT, since that would be inconsistent between source categories).

4.4.12. Use MOVES for on-road mobile sources

Extra files are needed for input to SMOKE when running with MOVES. These files are:

- MEPROC: Mobile emission processes

- MCXREF: Mobile reference county cross-reference file

- MFMREF: Mobile fuel month file

- MRCLIST: List of MOVES Lookup Tables for **Movesmrg**

- RUNCTL: Runcontrol file

The `MEPROC`, `MCXREF`, and `MFMREF` files are installed in the `$MBDAT` directory for the default SMOKE configuration.

The subsections below describe how adjusting these files will allow you to perform a variety of changes to the on-road mobile processing.

4.4.12.1. Coordinating with other input files

The `MEPROC` file must be coordinated with the `INVTABLE`, `GSPRO`, and `GSREF`. The `MCXREF` and `MFMREF` must be coordinated with the inventory. The following list indicates all of the things that you must check for when creating any of these files.

- The `MEPROC` file must have all of the pollutants (e.g., CO, NOX, and so on) as an input file to **Movesmrg**.

- If a pollutant that you want to model is not in the `MEPROC` file, it will not be modeled by SMOKE correctly. All pollutants that you wish to include in the outputs must be in the `MEPROC` file.

- All pollutants in the `MEPROC` file must also be in the `INVTABLE` file with Keep = Y, including the `INVTABLE` limitation that the file pollutant names are limited to 11 characters for on-road mobile pollutants.

- All pollutants in the `MEPROC` file must be available in SMOKE-ready MOVES lookup tables.

- All process-pollutant combinations in the `MEPROC` file must also be included in the `GSREF` file. The format for the "pollutant" entry in the `GSREF` is <emission process>__<pollutant>. Note that the joiner between <emission process> and <pollutant> is a double underscore. **A single underscore will prevent the processing from working properly.** For example, running exhaust carbon monoxide must be included as EXR__CO.

- When integrating toxics HAPs and criteria TOG and using MOVES, the NONHAPTOG definition in the `INVTABLE` file and `GSPRO` files must be consistent.

- The `MCXREF` file should assign reference counties for all counties for which you provide VMT data in the inventory.

4.4.12.2. Changing reference counties

Reference counties have previously been explained in Section 2.8.4.6, "Reference Counties" [41]. The `MCXREF` file assigns reference counties for each county in your domain. To change the reference counties, this file should be edited to assign different reference counties to each county in the inventory. After editing this file, you must make sure that no counties in the inventory have been left out of the file.

4.4.12.3. Changing fuel month for reference counties

Reference fuel month have previously been explained in Section 2.8.4.7, "Reference Fuel month" [42]. The `MFMREF` file assigns fuel month for each reference county in your domain. To change the fuel month, this file should be edited to assign a different fuel month to each reference county in the inventory. User must meet the criteria to share the same fuel month between reference counties.

4.4.13. Change temporal processing

Changing the temporal aspect of SMOKE processing is one of the critical changes that users must make. One of the most basic changes that you can make to SMOKE is the time period for which SMOKE will be run. In this section, we will explain how to make the following changes to your SMOKE processing:

- Section 4.4.13.1, "Changing temporal cross-reference and profiles" [161]

- Section 4.4.13.2, "Setting normalization of weekly profiles" [162]

- Section 4.4.13.3, "Changing the episode" [162]

- Section 4.4.13.4, "Changing the dates, times, and duration of model-ready SMOKE emissions files" [163]

- Section 4.4.13.5, "Setting non-sequential processing dates" [164]

- Section 4.4.13.6, "Changing holiday dates" [165]

- Section 4.4.13.7, "Use uniform temporal profiles for all sources" [165]

4.4.13.1. Changing temporal cross-reference and profiles

To change the temporal cross-reference and profiles the following steps must be taken.

1. Determine which days of the week will have distinct diurnal profiles.

   SMOKE provides the capability for optionally assigning different diurnal profiles for different days of the week. The file format for the ATPRO, MTPRO, and PTPRO file explains the different sections of the file that you can use to assign these different profiles. Before you add temporal profiles, you must decide which approach you will use for the new profiles:

   - Different diurnal profiles for weekdays and weekends (using the /DIURNAL WEEKDAY/ and /DIURNAL WEEKEND/ packets).

   - Different diurnal profiles for each day of the week (using the /DIURNAL <DAYNAME>/ packets, where <DAYNAME> = the day of the week (MONDAY, TUESDAY, etc.).

2. Ensure that different diurnal profiles for different days of the week will match across the days.

   When setting up new diurnal temporal profiles, it is important is ensure that the same temporal profile codes are used across the packets for the diurnal profiles because as source can only be assigned a single code for all of its diurnal profiles. For example, if a source were to use diurnal profile code 50, that code must be included in all of the diurnal packets and all instances of code 50 must be for that source. In other words, it is not possible to assign code 50 on Mondays and code 60 on Tuesdays.

3. Edit the ATPRO, MTPRO, or PTPRO file to include new diurnal, weekly, or monthly profiles.

   The file format for the ATPRO, MTPRO, and PTPRO files is in Chapter 8, *SMOKE Input Files* [345]. To add new diurnal profiles, you must edit these files in a text editor and the new profiles added. It is possible to use the same file instead of three separate files; in this case, just configure the Assigns file to use the same file name for all three logical file names. This has been done in the default configuration.

4. Edit the ATREF, MTREF, or PTREF file to add entries or change assignments.

The file format for the ATREF, MTREF, and PTREF files is in Chapter 8, *SMOKE Input Files* [345]. To add or change temporal profile assignments, you must edit these files in a text editor and select the source characteristics that will be associated with a diurnal, weekly, and monthly profile. The example file uses primarily SCCs to assign temporal profiles, but the file format supports other fields to make more specific assignments by county, pollutant, link ID, plant ID, or other point-source characteristics.

5. Ensure ATREF, MTREF, and PTREF have entries for all sources in the inventory.

Once you have edited the file, you will need to check that all sources are being assigned profiles or that the default temporal profile is acceptable for sources that do not have more specific assignments. The easiest way to check that the files are acceptable to you is to run it through the **Temporal** program using the REPORT_DEFAULTS option set to Y. With this option, the **Temporal** log file will report all sources that have received the default temporal profile. You can review the log file to determine if this default is acceptable to you. Additionally, you can run **Smkreport** (after **Temporal** has been run) with the following instructions in the REPCONFIG input file to get a report with the temporal profile codes that **Temporal** assigned to each source. The instructions are:
  • BY DIUCODE
  • BY WEKCODE
  • BY MONCODE
  • BY SOURCE

4.4.13.2. Setting normalization of weekly profiles

To set normalization of weekly profiles, the following steps must be taken.

1. Determine whether you are using an annual or average-day inventory.

   You are using an annual inventory if you have SMK_AVEDAY_YN is set to N in your run scripts.

   You are using an average-day inventory if you have SMK_AVE_DAY_YN set to Y in your run scripts.

2. If using an annual inventory, use weekly normalization.

   To do this, make sure the WKDAY_NORMALIZE setting is set to N in your run script when you run **Smkinven**.

3. If using an average-day inventory, determine whether it is an average weekday or not, and set the weekly normalization accordingly.

   If your average-day inventory is an average that excludes weekend emissions, then you need weekday normalization of your weekly profiles. To do this, make sure the WKDAY_NORMALIZE setting is set to Y in your run script when you run **Smkinven**.

   If the values of your average-day inventory include weekend emissions, then you do not need weekday normalization and the WKDAY_NORMALIZE setting in your run scripts should be set to N when you run **Smkinven**.

4.4.13.3. Changing the episode

The episode definition is the period for which you will generate SMOKE outputs. To change the episode dates, the following steps must be taken. In addition, the episode dates must be consistent with the run length settings, described in the next subsection.

1. Choose an episode within a single calendar year.

2. Set the `EPI_STDATE` to the Julian start date of your episode.

   In general, this setting is the same as the `G_STDATE` setting in the Assigns file. As will be described in the next subsection, `G_STDATE` changes with each modeling period within the episode (e.g., modeling with 5-day periods), but the `EPI_STDATE` setting is the same for the entire SMOKE run.

   The `EPI_STDATE` setting is also generally consistent with the base model year set by the `YEAR` environment variable, which in turn is consistent with the year of data in the input inventory files.

3. Set the `EPI_STTIME` to the start time that you wish to use for each model-ready emissions file created in your SMOKE run. This value is usually 0.

4. Set the `EPI_RUNLEN` to the number of hours in the episode.

   Do not add an hour to this duration as with the `G_RUNLEN` setting described in the next subsection. The format for this setting allows up to four hours in the format (i.e., HHHHMMSS).

5. Set the `EPI_NDAY` to the number of days of the episode.

   The minimum value is 1 and the maximum value is 365 (366 for leap years).

6. Ensure that all time dependent input data are available for the run period.

   The critical time-dependent data that you must ensure are available for the episode duration are:

   • Meteorology data (see Section 4.4.16, "Change meteorology data" [167])

   • Day- and hour-specific point source data (including any precomputed plume rise data)

4.4.13.4. Changing the dates, times, and duration of model-ready SMOKE emissions files

The run length is the time period for which each program that uses temporal process will be run. Sometimes these periods are called the "chunks" of processing, as in the episode is being processed in "chunks". This period will determine the length of all of the hourly SMOKE intermediate files including ATMP, MTMP, PTMP, PLAY, and all model-ready outputs from **Smkmerge**. To change the run length, the following steps must be taken.

1. Set the `G_STDATE` setting the same as your `EPI_STDATE`.

   The `G_STDATE` setting in the Assigns file is the starting date of the first SMOKE model-ready emissions file. The `G_STDATE_ADVANCE` script variable is set in the scripts so that the scripts will automatically reset the `G_STDATE` variable as need to run SMOKE for each period in the episode. The length of those periods is set by the `G_RUNLEN` variable and the number of periods is `EPI_RUNLEN` divided by `G_RUNLEN`.

2. Set the `G_STTIME` setting the same as the `EPI_STTIME`. This value is the start time that you wish to use for each model-ready emissions file created in your SMOKE run. This value is usually 0.

   Generally, the start time must also be consistent with the start time of the other inputs to the air quality model, such as the meteorology data. For example, if the meteorology data start at 12 Greenwich Mean Time (GMT) and the air quality model will also start at that time, then the emissions files will also need to start at 12 GMT. This can be achieved by setting `G_STTIME` and `EPI_STTIME` to 120000 (HHMMSS format).

3. Set the `G_TSTEP` setting to 10000 for 1 hour (HHMMSS format). In version 2 of SMOKE and earlier, this is the only setting that is supported by the code.

4. Set the `G_RUNLEN` to the period run length for each model-ready SMOKE emissions file.

    The run length can be set to as many hours as your air quality model can support for a single input file, as long as your files do not surpass a 2 GB file size limit, which are be imposed by 32-bit UNIX operating systems and by the NetCDF library on which SMOKE is based. Generally, the air quality models can accept files for as long as the user wishes to provide, so the limit is therefore the file sizes, which can be determined by trial and error. The conservative approach is to create emissions files for 1 day at a time.

    The air quality models require an extra hour of emissions data for each file. Therefore the number of hours for the `G_RUNLEN` setting is computed as:

    (number of days per file) x 24 + 1

    The `G_RUNLEN` setting must be in format of HHHHMMSS where HHHH is a four-digit number of hours, MM is the minutes (must be 00), and SS is the seconds (must also be 00).

5. Set the `ESDATE` setting to the first date of your episode in YYYYMMDD format.

    Recall that the `ESDATE` setting is for naming files. In the Assigns file, the value of this setting should be the starting date for the first emissions file to be created by the SMOKE run. The default SMOKE script configuration will automatically update `ESDATE` for subsequent run periods.

6. Set the `NDAYS` setting to the number of days represented by `G_RUNLEN`.

    For example, if `G_RUNLEN` is 490000 (2 days x 24 + 1 = 49 hours), then `NDAYS` should be 2.

4.4.13.5. Setting non-sequential processing dates

The `PROCDATES` option is specifically designed for use on daily and/or hourly inventory data with **Temporal**. `PROCDATES` is not yet incorporated within other SMOKE core programs, such as **Grdmat**, **Smkmerge** or **Smkreport**. For creating model-ready outputs and QA reports, the users need to be sure that the episode dates (`EPI_STDATE` and `G_STDATE`) in the assigns file are listed in `PROCDATES`

If `PROCDATES` is not assigned, the defaults are `G_STDATE`, `G_STTIME` and `G_RUNLEN`.

The following steps must be taken to set non-sequential processing dates.

Area/Non-road/Point Sources

1. Determine whether you need to use `PROCDATES`.

2. If you are using `PROCDATES`, you need to set the environment variable by uncommenting `PROCDATES` in the assigns file.

3. Set the environment variable `PROCDATES` to the file procdates.txt located in `GE_DAT`. (i.e. setenv `PROCDATES $GE_DAT/procdates.txt`.

Mobile Source

Temporal is responsible to convert annual/monthly to hourly VMT. **Movesmrg** is responsible for combining hourly emissions factors calculated by MOVES with hourly VMT data to generate hourly on-road mobile emissions.

1. Determine whether you need to use `PROCDATES`.

2. If you are using `PROCDATES`, you need to set the environment variable by uncommenting `PROCDATES` in the assigns file.

3. Set the environment variable `PROCDATES` to the file procdates.txt located in `GE_DAT`. (i.e. setenv `PROCDATES $GE_DAT/procdates.txt`.

NOTE: Since the **Temporal** dynamically create names for the output files, two new environment variables `[A|M|P]TMPNAME` and `[A|M|P]TSUPNAME` are used to set the directory and file prefix for naming the output files `[A|M|P]TMP` and `[A|M|P]TSUP`. The files are named using the starting date of each time period.For example, if `ATMPNAME` is set to /data/ntmp.nctox., then the `ATMP` file for a given time period will be put in the data directory and named ntmp.nctox.start date.ncf.

4.4.13.6. Changing holiday dates

As previously explain in Section 2.10.5, "Holiday processing" [67], SMOKE can model holidays as special days. The holidays that are relevant are those holidays for which anthropogenic activities are significantly altered, since its those activities that determine the appropriate weight to give the emissions on each day of the week and the diurnal temporal patterns for those days. For example, the nearly all workers in the U.S. have a holiday from their employment on Thanksgiving day, and the patterns of industrial and motor-vehicle activities are substantially altered resulting in different emissions patterns on those days. The Valentines Day holiday, however, does not affect such patterns because employers do not allow workers a day off on that day.

The `HOLIDAYS` file controls which days SMOKE will treat as holidays and which day of the week SMOKE will emulate for that day. The file must be created for each year, since some holiday dates are different each year. Although the format of the file suggests that the holidays can be modeled differently for different counties in the inventory, this is not the case - the format is provided for a future update to SMOKE, which has not yet been implemented.

To change the holiday dates that are modeled by SMOKE, all that is required is changing the `HOLIDAYS` file to include the year and holidays in which you are interested, and select a day of the week to use for modeling that holiday. If you do not wish SMOKE to treat a holiday any differently, then it should be removed from the file. If you wish no days to be modeled as holidays, we recommend including at least one date in the file, and assigning its actual day of the week, to prevent SMOKE from getting confused.

4.4.13.7. Use uniform temporal profiles for all sources

SMOKE includes a little-used option to allow SMOKE to ignore all temporal profile information from the temporal cross-reference and temporal profiles. In this case, the emissions will be constant for all hours of all days, except for a slight variation from month to month to account for the different number of days in each month. This option still requires the temporal cross-reference and temporal profile input files, but they are ignored. To use this capability in SMOKE, set the `UNIFORM_TPROF_YN` setting in the run script to Y.

4.4.14. Use Elevpoint for selecting elevated and PinG sources

The **Elevpoint** program can select elevated and/or PinG sources as needed for some air quality models and configurations. You must use the **Elevpoint** program in the following cases:

• You are running UAM, REMSAD, or CAM$_X$ including elevated point sources and/or PinG sources.

- You are running CMAQ and want to select specific elevated sources or want to model with the PinG capability.

- You are running MAQSIP and want to select specific elevated sources.

The following steps must be taken to use **Elevpoint**:

1. Create a PELVCONFIG file.

   The PELVCONFIG file is the **Elevpoint** input file that controls what sources will be selected as elevated and/or PinG. Chapter 8, *SMOKE Input Files* [345] explains the file format for this file and how to use this format. A example PELVCONFIG file provided with SMOKE (in the nctox case PTDAT directory) contains additional instructions on setting up this file.

2. Update Assigns file to use the correct name of the PELVCONFIG file.

3. Set RUN_ELEVPOINT to Y in point source run script.

4. If using PinG sources, set SMK_PING_METHOD to 1 in point source run script (otherwise, set to 0).

5. If selecting specific elevated sources, set SMK_ELEV_METHOD to 1 in point source run script (otherwise, set to 0).

6. See model-specific settings in Section 4.4.19, "Model for CMAQ and MAQSIP" [170] and Section 4.4.20, "Model for UAM, REMSAD, and CAM$_X$" [172].

4.4.15. Use BEIS3

The steps you need to take to use BEIS3 for your modeling case are:

1. Ensure that the land use file has been created and properly installed

   See Section 4.4.9.2, "Installing or changing spatial input files" [146], step 4) for more information on creating and installing the land use data.

2. Ensure that the meteorology data are correct

   See Section 4.4.9.2, "Installing or changing spatial input files" [146], step 5) and Section 4.4.16, "Change meteorology data" [167] for more information on creating and installing the meteorology data.

3. Ensure that BEIS3 emission factors are in place.

   The B3FAC file is installed by SMOKE and does not need to be changed, so it is very unlikely that this file will not be in its GE_DAT location. The example Assigns file sets the logical file names for this file, which should not be changed.

4. Setup your BEIS3 run script

   To set up your BEIS3 run, copy the example BEIS3 run script listed in Section 4.2.3, "Example script files and their purposes" [120] and reset the script settings. The settings that you may need to reset are as follows (see Chapter 6, *SMOKE Core Programs* [211] for all program options):

   - RUN_METSCAN: Set to Y if you need to use the BIOSEASON file (see Section 4.4.9.2, "Installing or changing spatial input files" [146], step 6).

- **Normbeis3** options

- **Tmpbeis3** options

- **Smkmerge** options - this program is used to create state/county totals and/or to change units for CMAQ from moles/hr (output by **Tmpbeis3**) to moles/s. Depending on the purpose for running **Smkmerge** in your script, adjust the **Smkmerge** options accordingly. Please note that you cannot create gridded outputs in moles/s and state/county totals in tons/day in a single run.

  If creating state/county totals, make sure the AREA_SURROGATE_NUM setting is set to the cell-area surrogate number in your BGPRO file.

- SPC_OVERRIDE: May be needed in some cases (see Section 4.2.4, "Script settings" [121])

4.4.16. Change meteorology data

To change the meteorology data, you must take the following steps:

1. Determine what format your air quality model will accept for meteorology data and convert data to I/O API format if needed.

   If you are modeling with CMAQ or MAQSIP, then the data will be in the I/O API format created by MCIP or MCPL that SMOKE can accept. If you are modeling with UAM, REMSAD, or CAM_X, then you must check to see which meteorology format the model accepts. Some versions of these models can use the same I/O API format as for CMAQ, but some cannot.

   If your meteorology data are not in the correct format, then you will need to convert the data. There is currently no utility provided with SMOKE to convert the meteorology format needed by some versions of UAM, REMSAD, and CAM_X to SMOKE input format.

2. Ensure required variables are included in meteorology data

3. Ensure meteorology data cover the correct time period

4. Ensure meteorology data are on the correct grid

   As described in Section 4.4.9.2, "Installing or changing spatial input files" [146] step 5, check that the meteorology grid and modeling grid are consistent.

5. Install meteorology data

   The meteorology data should be installed in the $METDAT directory, as explained in Chapter 3, *SMOKE Directory Structure* [97]. The names of the files are up to you.

6. Ensure that the meteorology data names match with the names in the Assigns file

   Edit the Assigns file and look for the $METDAT variable. Ensure that it is consistent with the actual location of the meteorology files. Also check the settings for GRID_CRO_2D, GRID_CRO_3D, MET_CRO_2D, MET_CRO_3D, and MET_DOT_3D and make sure that the naming scheme for these files is consistent with your file names.

4.4.17. Plume Rise Calculation for Fires

Starting with version 2.2, SMOKE contains an approach for computing plume rise for wildfire/prescribed fires using the Bluesky Pouliot-Godowitch plume rise algorithm (http://www.epa.gov/ttn/chief/confer-

ence/ei14/session12/pouliot.pdf). In SMOKE version 2.2, SMOKE can compute hourly plume rise from precomputed hourly heat flux, fuel loading and area burned data using the Bluesky CONSUME model. The SMOKE preprocessor **Bluesky2Inv** that converts the output from the CONSUME model into SMOKE-readable annual and daily inventory files (Section 5.3.3, "**BlueSky2Inv**" [190]).

Starting with SMOKE version 2.3, SMOKE can compute hourly plume rise from an intenally computed heat flux using area burned and fuel loading data. This approach allows SMOKE to compute hourly heat flux based upon daily area burned, fuel loading, material burned and a constant default heat content (8000 BTU/lb) (Equation 1). The daily heat flux and emissions are temporally allocated to hourly values based upon the duration of the fire (begining and ending hours). However, if there is a precomputed heat flux available as a daily inventory, **Smkinven** will not internally compute the heat flux and drop the begin and end hours of fire.

1. Heat Flux (BTU/hr) = [Area Burned (acre/day) x Fuel Loading (tons/acre) x Heat Content (BTU/lb) x (2000lb/ton)] / Duration of fire (hr/day)

The SMOKE plume rise requires two ORL formatted files for wildfire/prescribed fires. The master #ORL FIRE file (Section 8.2.8.3, "ORL FIRE Format" [380]) contains the fire characteristics including country/state/county code, fire idenfication code, geographic coordinate, fire name, heat content, material burned and NFDRS code. The day-specific #ORL FIREEMIS file ("Section 8.2.6.2, "ORL FIRE Emission Format" [368]") includes daily pollutant emissions, area burned, fuel loading, optional heat flux, and fire begining and ending time. Starting with SMOKE version 3.7, **Smkinven** can process these master and daily fire inventories with precomputed heat flux using current point FF10 formats, #FF10_POINT file ("Section 8.2.8.1, "FF10 Format" [372]") and #FF10_DAILY_FF10 file ("Section 8.2.6.1, " FF10 Format" [367]")

Regardless whether the heat flux for wildfires is precomputed or internally computed, the fire-specific plume rise algorithm converts the heat flux to a bouyancy flux for use in the plume rise calculation. See more detail at Section 6.6.1.1, "Plume rise calculation for fires" [235]

NOTE: SMOKE requires that the fire inventory pollutant names are copied to the Chemical Abstracts Service (CAS) number column in the inventory table (`INVTABLE`) file (See Section 8.10.4, "`INVTABLE`: Inventory table" [450]).

1. To compute plume rise for fires using SMOKE, following these steps. Add the fire inventory variables to the `INVTABLE` file:

```
HFLUX           HFLUX                    Y      1 N N N N    0 tons/yr
HEATCONTENT     HEATCONTENT              Y      1 N N N N    0 tons/yr
FUEL_LOAD       FUEL_LOAD                Y      1 N N N N    0 tons/yr
ACRESBURNED     ACRESBURNED              Y      1 N N N N    0 tons/yr
BEGHOUR         BEGHOUR                  Y      1 N N N N    0 tons/yr
ENDHOUR         ENDHOUR                  Y      1 N N N N    0 tons/yr
```

2. Run **Smkinven** with these settings,

    • Set `RUN_SMKINVEN` to Y

    • Set `IMPORT_AVEINV_YN` to Y

    • Set `CHECK_STACKS_YN` to N

3. Do not run **Elevpoint**,

    • Set `RUN_ELEVPOINT` to N

- • Set `SMK_ELEV_METHOD` to 0

- • Set `SMK_PING_METHOD` to 0

4. Run **Laypoint** with these settings,

- • Set `RUN_LAYPOINT` to Y

- • Set `FIRE_PLUME_YN` to Y

- • Set `HOURLY_FIRE_YN` to Y

- • Set `SMK_SPECELEV_YN` to N

5. Run **Smkmerge** with theses settings,

- • Set `RUN_SMKMERGE` to Y

- • Set `SMK_ASCIIELEV_YN` to N

6. Set these multiple program control settings.

- • Set `DAY_SPECIFIC_YN` to Y

- • Set `EXPLICIT_PLUMES_YN` to N

- • Set `SMK_PING_METHOD` to 0

- • Set `QA_TYPE` to part2

4.4.18. In-line Plume Rise Calculation in CMAQ model (as of version 4.7)

Staring with version 2.5, SMOKE supports the calculation of plume rise for point sources in CMAQ known as an in-line plume rise calculation. SMOKE is used to select the point sources to reduce the CMAQ plume rise treatment. In this section, we give a basic outline of how to run SMOKE to output emissions for the in-line plume rise processing in CMAQ. To prepare emissions for the CMAQ in-line plume rise calculation follow these steps:

1. For fires, the following two lines need to be added to the `GSPRO` file (See Section 8.5.2, "GSPRO: Speciation profile file" [390]):

    `"0000";"HFLUX";"HFLUX";1;1;1`

    `"0000";"ACRESBURNED";"ACRESBURNED";1;1;1`

2. Set these variables in the SMOKE run scripts.

- • Set `RUN_LAYPOINT` to N

- • Set `RUN_ELEVPOINT` to Y

- • Set `SMK_ELEV_METHOD` to 2 (See Section 6.3.3.2, "Input Environment Variables" [226])

- • Set `SMK_PING_METHOD` to 0

- Set `MRG_LAYERS_YN` to N

3. **Elevpoint** to select the sources to receive the CMAQ in-line plume rise treatment. Create a `PELVCON-FIG` file to group sources with nearly identical stack parameters to reduce the total number of plume rise calculations performed by the CMAQ in-line plume rise module. Non-fire sources with an analytical plume rise height greater than 38 meters are typically selected for in-line plume rise processing. The `PELVCONFIG` file used for CMAQ in-line processing of non-fire sources should be similar to the following example:

SMK_SOURCE P

/SPECIFY ELEV GROUP/

HT +/- 0.001 AND DM +/- 0.001 AND TK +/- 0.001 AND VE +/- 0.001

/END/

/SPECIFY ELEV/

RISE >= 38

/END/

For point source fires, such as the EPA ptfire sector, the following is an example of a `PELVCONFIG` file:

SMK_SOURCE P

/SPECIFY ELEV GROUP/

HT+/- 1.0

/END/

/SPECIFY ELEV/

PM2_5 > 0

/END/

4.4.19. Model for CMAQ and MAQSIP

In this section, we attempt to give a basic outline of the considerations for script reconfiguration needed for SMOKE output for the CMAQ and MAQSIP models. All models require **Smkmerge** to be configured for outputting gridded data, with speciation, and with hourly emissions. The following configurations are needed when modeling for CMAQ and MAQSIP.

1. Make sure the chemical speciation is set for one of the CMAQ or MAQSIP chemical mechanisms.

   This check involves the `SPC` setting, explained in Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150].

2. Set the basic settings for CMAQ or MAQSIP.

   - Set `RUN_LAYPOINT` to Y

- Set `REP_LAYER_MAX` to the number of the layer you expect should be the maximum plume rise layer (the layer that includes 2000 meters).

- Set `MRG_TEMPORAL_YN` to Y

- Set `MRG_SPCMAT_YN` to Y

- Set `MRG_LAYERS_YN` to Y

- Set `MRG_GRDOUT_YN` to Y

- Set `SMK_ASCIIELEV_YN` to N

- Set `SMK_EMLAYS` to the number of layers you need for your emissions files (usually up to the layer that includes 2000 meters, unless you are using explicit plume rise, in which case the layer that includes the maximum plume height of the precomputed plume top).

3. Set the correct output units in the runs scripts.

   CMAQ expects output units of moles/s. For biogenic processing, this requires the **Smkmerge** program to be run to convert the moles/hr units output by **Tmpbio** or **Tmpbeis3** to moles/s.

   MAQSIP expects output units of moles/hr.

   Set the `MRG_GRDOUT_UNIT` setting in the run script to either moles/s (for CMAQ) or moles/hr (for MAQSIP). For biogenics processing, make sure `RUN_SMKMERGE` is set to Y.

4. Configure approach for plume rise for standard point sources, including optional PinG sources for CMAQ.

   The point-source run script should have the following settings in each of the following situations.

   a. Calculate plume rise for all point sources.

      - Set `RUN_ELEVPOINT` to N

      - Set `HOUR_PLUMEDATA_YN` to N

      - Set `SMK_SPECELEV_YN` to N

      - Set `EXPLICIT_PLUMES_YN` to N

      - Set `SMK_PING_METHOD` to N

      - Set `SMK_SPECELEV_YN` to N

   b. Select specific elevated sources for CMAQ or MAQSIP and calculate plume rise only for those.

      Use the same settings as in part (a) above, except:

      - Set `RUN_ELEVPOINT` to Y (See Section 4.4.14, "Use **Elevpoint** for selecting elevated and PinG sources" [165] for more information about running **Elevpoint** to select elevated sources)

      - Set `SMK_ELEV_METHOD` to 1

      - Set `SMK_SPECELEV_YN` to Y

   c. Select PinG sources for CMAQ.

Use the same settings in part (a) above, except:

- Set RUN_ELEVPOINT to Y (See Section 4.4.14, "Use **Elevpoint** for selecting elevated and PinG sources" [165] for more information about running **Elevpoint** to select PinG sources)

- Set SMK_PING_METHOD to 1

5. Configure approach for plume rise for explicit plume rise sources.

Explicit plume rise sources (i.e., sources for which plume rise has been computed outside of SMOKE) require additional precomputed plume rise information, previously described for inventory data import in Section 4.4.2.5, "Day-specific and hour-specific point and fire inventories" [136]. In addition to these data for CMAQ or MAQSIP modeling, the following settings must be made in the run scripts. These settings are in addition to the other configurations you need for the other point sources described above. If the only point sources in your inventory are explicit plume rise sources, then use the settings described in steps 3 and 5a above.

- Set HOUR_PLUMEDATA_YN to Y

4.4.20. Model for UAM, REMSAD, and CAM$_X$

1. Make sure the chemical speciation is set for one of the UAM, REMSAD, or CAM$_X$ chemical mechanisms.

This check involves the SPC setting, explained in Section 4.4.10, "Use a different speciation mechanism or change speciation inputs" [150]. Please note that the files and setup needed for CAM$_X$-PM are not available in SMOKE v2.1.

2. Set the basic settings for UAM, REMSAD, and CAM$_X$

In these settings, we will configure the system to select elevated point sources using the **Elevpoint** program, but not PinG sources.

- Set RUN_LAYPOINT to N

- Set RUN_ELEVPOINT to Y (See Section 4.4.14, "Use **Elevpoint** for selecting elevated and PinG sources" [165] for more information about running **Elevpoint** to select elevated sources)

- Set SMK_ELEV_METHOD to 1

- Set MRG_TEMPORAL_YN to Y

- Set MRG_SPCMAT_YN to Y

- Set MRG_LAYERS_YN to N

- Set MRG_GRDOUT_YN to Y

- Set SMK_ASCIIELEV_YN to Y

- Set MRG_GRDOUT_UNIT to moles/hr

- Set EXPLICIT_PLUMES_YN to N

- Set HOUR_PLUMEDATA_YN to N

- Set `SMK_SPECELEV_YN` to Y

3. Optionally configure for PinG sources.

   In addition to different instructions in the `REPCONFIG` file, the point-source run script should have the following settings in each of the following situations.

   - Set SMK_PING_METHOD to 1

4. Configure approach for plume rise for explicit plume rise sources.

   Explicit plume rise sources (i.e., sources for which plume rise has been computed outside of SMOKE) require additional precomputed plume rise information, previously described for inventory data import in Section 4.4.2.5, "Day-specific and hour-specific point and fire inventories" [136]. In addition to these data for UAM, REMSAD, and CAM$_X$ modeling, the following settings must be made in the run scripts. These settings are in addition to the other configurations you need for the other point sources described above. If the only point sources in your inventory are explicit plume rise sources, then use the settings described in steps 2 and 3 above.

   - Set `RUN_LAYPOINT` to Y

   - Set `EXPLICIT_PLUMES_YN` to Y

   - Set `HOUR_PLUMEDATA_YN` to Y

4.4.21. Determine if a run worked correctly and troubleshoot

Determining if your run worked correctly and troubleshooting are described in detail in Chapter 7, *SMOKE Quality Assurance* [305]. We refer the reader to that chapter for explanations of how to perform these steps.

1. Resolve all errors and warnings in all log files.

2. Compare SMOKE output emissions to inventory total emissions at each stage of SMOKE processing to ensure emissions have not been erroneously lost or added.

3. Review and analyze SMOKE reports that indicate the speciation profiles, temporal profiles, and spatial surrogate codes assigned to each SCC in the inventory.

4. Examine your inventory for interregional inconsistencies.

5. Visually examine your model-ready emissions to search for possible errors.

4.4.22. Change reports

SMOKE provides the **Smkreport** tool as a very flexible tool to generate reports on your inventory and emissions modeling case. This program is documented in detail in Chapter 7, *SMOKE Quality Assurance* [305]. Here, we provide the information that you need to reconfigure your scripts to use different reporting. The `QA_TYPE` and `QA_LABEL` environment variables introduced in Section 4.2.4, "Script settings" [121] are clarified here.

1. Customize the names of the output report files and **Smkreport** log files.

The example SMOKE run scripts will name the log files and report output files with preset names. If you set the `QA_LABEL` setting, the reports and log file names will then include the value of that setting. For example, if the default report name were `a.state.nei99.rpt` and `QA_LABEL` were set to "run1", the new report name would be `a.state.nei99.run1.rpt`. The "run1" in `QA_LABEL` is inserted before the final "rpt" extension.

2. Select from one of the default settings for `QA_TYPE`

The default `QA_TYPE` settings allow users to run **Smkreport** with the default `REPCONFIG` files. These files are located in the `$OTHER` directory, documented in Chapter 3, *SMOKE Directory Structure* [97]. Each source category has its own valid `QA_TYPE` settings, and each of these settings causes certain reports to be created. Table 4.5, "`QA_TYPE` settings with associated source categories and default `REPCONFIG` files" [174] indicates which `QA_TYPE` setting can be used in which case, and the `REPCONFIG` files that will be used.

Table 4.5. `QA_TYPE` settings with associated source categories and default `REPCONFIG` files

`QA_TYPE`	Source Categories	`REPCONFIG` file from `$SMKDAT/inventory/nc-tox/other`
part1 (RUN_PART1 must be set to Y)	area nonroad mobile point	`repconfig.ar.inv.txt` `repconfig.nr.inv.txt` `repconfig.mb.inv.txt` `repconfig.pt.inv.txt`
part2 (RUN_PART2 must be set to Y)	area nonroad mobile point	`repconfig.ar.temporal.txt` `repconfig.nr.temporal.txt` `repconfig.mb.temporal.txt` `repconfig.pt.temporal.txt`
part3 (RUN_PART3 must be set to Y)	point	`repconfig.pt.elev.txt`
part4 (RUN_PART4 must be set to Y)	point	`repconfig.pt.lfrac.txt`
all	area nonroad mobile point	All of the above reports will be generated, provided that the `RUN_PART(1-4)` settings are set as listed in column 1 of this table.
custom	area nonroad mobile point	See step (3) below.

The helper script `qa_run.csh` is responsible for selecting the correct `REPCONFIG` file based on the `QA_TYPE` setting, and these settings need not be changed. If users wish to implement a different configuration, the "custom" value of `QA_TYPE` should be able to meet those needs without any changes to `qa_run.csh`. Copying the `REPCONFIG` files from the example files to your case and changing them changes the default reports that are generated by SMOKE for your case.

Users can examine these `REPCONFIG` files along with the documentation in Chapter 7, *SMOKE Quality Assurance* [305] to determine the reports that will be generated by each. These reports are also

generated during the SMOKE installation and test case run, and can provide examples for understanding which reports will be written.

3. Using the "custom" setting

   If QA_TYPE is set to "custom", then the qa_run.csh script expects that users will set the correct REPORT* environment variables in the main body of their run script. The REPORT* environment variables are typically used as the logical file names for **Smkreport**, those environment variables used are themselves configurable in the REPCONFIG file. The physical file names can also be set in the REPCONFIG file directly, instead of relying on the scripts to contain environment variable settings. These options are explained in greater detail in Chapter 7, *SMOKE Quality Assurance* [305].

4.4.23. Output source apportionment data

Source apportionment in SMOKE consists of grouping sources by characteristics (i.e. FIPS code, SCC, and point source information) and then tagging the emissions from those groupings for further analysis in the air quality model. The SMOKE programs **Movesmrg** and **Smkmerge** can optionally output the emissions in special files used for later source apportionment processing.

The following steps outline the process for creating the source apportionment output files:

1. Define source groups using the SOURCE_GROUPS file

   The format for the SOURCE_GROUPS file is described in Section 8.10.13, "SOURCE_GROUPS: Source apportionment groups" [458]. When assigning sources to groups, SMOKE will match each source to the most specific line in the SOURCE_GROUPS file. The hierarchy of assignment priority is listed below.
   a. Country/state/county code, SCC, plant ID, point ID [point sources only]
   b. Country/state/county code, SCC, plant ID [point sources only]
   c. Country/state/county code, plant ID, point ID [point sources only]
   d. Country/state/county code, plant ID [point sources only]
   e. Country/state/county code, SCC [non-biogenic sources only]
   f. Country/state code, SCC [non-biogenic sources only]
   g. SCC [non-biogenic sources only]
   h. Country/state/county code
   i. Country/state code
   j. Default group (country/state/county code = 0, SCC = 0)

2. Define locations for the output files SRCGROUPS_OUT (source group information), SGINLNTS_L (emissions data), and SRCGRP_REPORT (source group report file)

3. For point source processing, use in-line plume rise calculation

   Review Section 4.4.18, "In-line Plume Rise Calculation in CMAQ model (as of version 4.7)" [169] for instructions on configuring **Elevpoint** and **Laypoint** for in-line emissions processing.

4. After completing other SMOKE processing, run **Movesmrg** (on-road mobile emissions using MOVES lookup tables) or **Smkmerge** with SMK_SRCGROUP_OUTPUT_YN set to Y

   When processing emissions using source apportionment, you can only run **Smkmerge** for a single source category (i.e. A, B, M, or P). For point source processing, **Smkmerge** cannot merge layers (MRG_LAYERS_YN must be N) and in-line emissions must be output (SMK_ELEV_METHOD must be 2). **Movesmrg** and **Smkmerge** will create both the source apportionment data files and the model-ready output files in a single run. For biogenics, if you are running **Smkmerge** for reporting purposes

(i.e. to create state and county totals), make sure to set the units correctly for the source apportionment output data, typically the same as the model-ready output units.

5. Review source grouping assignments

    After running **Movesmrg** or **Smkmerge**, the SRCGRP_REPORT file will list each source in the inventory and the group it was assigned to. You can review this report to verify that the sources were grouped as desired.

Chapter 5. SMOKE Utility Tools

Table of Contents

5.1. Overview

The utility tools are divided into two sections: MOVES Utility Scripts and SMOKE Utility Programs

5.2. MOVES Utility Scripts

5.2.1. Overview

The MOVES utility scripts are designed to assist the user by automating the proper use of using MOVES emission rate calculations for the purpose of estimating mobile-source emissions for air quality modeling. The MOVES default database has over 100 different tables that store lookup/reference information, conversions/adjustment factors, emissions data, and activity data. MOVES uses MySQL databases to store user inputs, intermediate results and final outputs. The MOVES utility scripts consist of 1) MOVES Driver script and 2) MOVES Post-processing scripts. These scripts are run after the meteorological data processor program (**Met4moves**) which prepares the spatially and temporally averaged temperatures and relative humidity data used as meteorological input conditions for MOVES and SMOKE using the Meteorology-Chemistry Interface Processor (MCIP) output files.

The MOVES Driver script (**Runspec_generator.pl**) creates data importer files and importer script and the MOVES batch run script which specifies the characteristic of the particular scenario to be modeled.

The MOVES Post-processing script (**Moves2smkEF.pl**) formats the MOVES emission rate lookup tables for SMOKE. The second MOVES Post-processing script (**gen_8digit_scc.pl**) generates an SCC mapping file used when importing activity data in **Smkinven**. After the MOVES utility scripts are completed, the SMOKE postprocessing program (**Movesmrg**) estimates emissions from mobile sources using the SMOKE formatted MOVES lookup tables (RPD, RPP, RPV, and RPH) output generated by the MOVES utility scripts and meteorology data from **Met4moves**.

Movesmrg creates hourly, gridded, speciated air quality model-ready input files. Information about key concepts related to the MOVES utility scripts including Reference Counties, Fuels Months, and Emissions Processes are covered in Section 2.8.4.6, "Reference Counties" [41] and Section 2.8.4.7, "Reference Fuel month" [42] and Section 2.8.4.9, " MOVES Emission Processes by Emissoin Rate Tables" [43]

5.2.2. MOVES Driver Script

5.2.2.1. Description

The MOVES Runspec script **Runspec_generator.pl** is written in Perl, generates the scripts and inputs needed to run MOVES for each reference county. The MOVES Model can be run at any of the three domains/scales: national, county or project. The **Runspec_generator.pl** is only applicable to the county-level

because this level of model detail is required by EPA for SIP and conformity analyses. For this scale, MOVES requires a MySQL input database containing local data for a single county. The **Runspec_generator.pl** is designed to prepare the MOVES importer and MOVES run scripts so that the fewest number of runs will produce all of the necessary emissions factors. In this approach, user can specify one or more groups of pollutants to model in the run control fil.

There are a total of four types of modeling pollutant groups: 1) Ozone precursors, 2) Toxics, 3) Particulate Matters, and 4) GreenHouse Gases (GHG) based on Table 2.12, "MOVES Pollutant Groups" [46]. The choice of pollutant group(s) determines what pollutants are included in the three emission rate lookup tables (RPD, RPV, and RPP) by MOVES.

5.2.2.1.1. Processing Order

Runspec_generator.pl run after **Met4moves**. It assembles the instructions for MOVES to create MySQL input databases from XML files and creates the importer.csh scripts and assembles run specification XML files to run MOVES for a necessary range of conditions (for each emissions process: rate-per-distance, rate-per-vehicle, and rate-per-profile) based on the **Met4moves** output temperature list by county-specific runspec.csh.

The **Runspec_generator.pl** uses following steps:

1. Reads the reference county min/max temperature and 24-hours temperature profiles output file from **Met4moves** for RPP mode. Find detail information from Section 9.8.1, "`MOVES_OUTFILE`" [470]

2. Reads the reference county specific min/max temperatures and average RH output file from **Met4moves** for RPD, RPV and RPH modes. Find detail information from Section 9.8.2, "`MOVES_RH_OUT-FILE`" [471]

3. Import the name of pollutant group to model from the run control file (contorl.in) based on Table 2.12, "MOVES Pollutant Groups" [46]

4. Reads the run control file (control.in) and reference county file (countyrep.in) created by user.

5. Assembles instructions for MOVES to create MySQL input databases from XML files (data importer).

6. Assembles runspec XML files to run MOVES for a wide range of conditions.

7. Generates the run-specific temperature and humidity comma-separated-values (CSV) format file.

8. Assembles a batch list of data importer files, runspec files, and also a list of the MySQL output database names to be postprocessed.

9. importer.csh and runspec.csh scripts are used to create the on-road operating rateperdistance table (RPD), off-network processes ratepervehicle table (RPV), and vapor venting off-network rateperprofile table (RPP).

10. Review log file of the MOVES, called runlog_*.txt.

5.2.2.1.2. Files, Environment Variables, and Run Syntax

5.2.2.1.2.1. Input Files

Table 5.1. Input Files

File Name	Format	Description
countyrep.in	ASCII	Reference county file containing the county/state/country FIPS codes and the full path and file names for the files containing MOVES data for AGE, FUELSUPPLY, FUELFORM(optional), IM, POP, SourceType, HPMSVMT, VMT for each reference county
control.in	ASCII	Run control file that user specifies fields for the Database Host, Project Name, Model year, day type (weekday or weekend), pollutant and the met4moves MET output file for the batch run of MOVES
MOVES_OUTFILE	ASCII	**Met4moves** output for each fuel month for the MOVES Driver script (minimum and maximum temperatures, and 24-hour temperature profiles for all reference counties by temperature bin)
MOVES_RH_OUTFILE	ASCII	**Met4moves** outputs fuel month-specific averaged, min, and max temperatures and averaged RH values in local time for all reference counties by temperature bin
sourceTypeAgeDistribution	ASCII	Age distribution by source type over 30 vehicle model years
fuelSupply	ASCII	list directed file used to define the fuel properties, their market shares and the months that the fuels are used for each reference county
fuelformulation (optional)	ASCII	list directed file used to define the fuel supply and the months that the fuels are used for each reference counties
IMCoverage	ASCII	list directed file used to define the inspection and maintanance data available for each reference counties
sourcetypeyear	ASCII	Vehicle population by MOVES source type
HPMSVTypeYear	ASCII	list directed file used to define the annual VMT by Highway Performance Monitoring System (HPMS) vehicle type
dummy_avgspeeddistribution	ASCII	Speed distribution by roadway type and vehicle class. MOVES dummy inputs files provided with tool package in OUTDIR directory
dummy_monthvmtfraction	ASCII	list directed file that allocates annual VMT to the individual months
dummy_dayvmtfraction	ASCII	file that allocates month VMT day type, weekday or weekend and the decimal fraction of the proportion of vehicle miles travelled for the weekday or weekend day type
dummy_hourvmtfraction	ASCII	list directed file that allocates the day-type VMT to individual hours of the day
dummy_roadtypedistribution	ASCII	list directed file that allocates annual VMT to five MOVES roadway types

5.2.2.1.2.2. Run Syntax
```
perl runspec_generator.pl control.in countyrep.in
```

5.2.2.1.2.3. Output Files

File Name	Format	Description
runspec.csh	ASCII	Batch or c-shell script to run required MOVES runs. Fine more information from Section 5.2.3.3, "MOVES Runspec batch script" [181]
importer.csh	ASCII	Batch or c-shell Script to create MySQL MOVES databases and import the data. Fine more information from Section 5.2.3.2, "MOVES importer script" [180]

InputDBlist	ASCII	List of MySQL database names to process. It will be used as an input to MOVES Post-processing script. See example below:
		
		`localhost`
		`/work/MOVES_Processing_Scripts/post_process`
		`13121_2008_1`

Data importer files	ASCII	data importer files (*_imp.xml)
Runspec files	ASCII	runspec files (*_mrs.xml)
zonemonthhour files	ASCII	MOVES-formatted zonemonthhour (*_zmh.csv) contains temperature and RH data from the Met4Moves Meteorological processor. Relative Humidity is a single value averaged over the time period selected by the user in Met4Moves and averaged over the entire group of inventory counties that map to a reference county.

5.2.3. Run MOVES Model

5.2.3.1. Overview

Automatically execute multiple MOVES runs based on automatically generated county-specific data importer script and the MOVES run script file by **Runspec_generator.pl**, which specifies the characteristic of the particular scenario to be modeled.

5.2.3.2. MOVES importer script

5.2.3.2.1. Description

The MOVES importer script "importer.[bat|csh]" can be written in Linux C-shell or Windows Batch script format depending on which option is chosen by the user in the **Runspec_generator.pl**. The importer script loads the county-specific data in the file and creates the MySQL input database containing local data for the reference county.

The output of the MOVES batch importer script includes:

- MySQL databases (*_in).

5.2.3.2.2. Processing Order

importer.csh should be run before runspec.csh. The importer.csh is written in Linux c-shell script format.

The importer.csh uses following steps:

1. The importer.csh contains all of the user selections needed to import data from the files into a MySQL MOVES database

2. Imports the data for each of the three different emissions processes: RPD, RPV, RPP.

3. Review log file of the MOVES, called importlog_*.txt.

5.2.3.2.3. Files, Environment Variables, and Run Syntax

5.2.3.2.3.1. Input Files

File Name	Format	Description
RV_[CountyRep]_[yearID]_[monthID]_[Temp-Bin]_imp.xml	XML	file specifies the zonemonthhour (*zmh.csv) files and other files that will be imported to MOVES for the Rate Per Vehicle Mode, note: TempBin varies, in 5 degree increments to cover the Temperature Range ie. [TempBin] = T10, T15,.. T105 for the temperature range [TempRange]= T10_105
RP_[CountyRep]_[yearID]_[monthID]_[Met-ProfID]_imp.xml	XML	file specifies the zonemonthhour (*zmh.csv) files that will be imported to MOVES for the Rate Per Process Mode
RD_[CountyRep]_[yearID]_[monthID]_[TempRange]_imp.xml	XML	importer file specifies the zonemonth-hour (*zmh.csv) files that will be imported to MOVES for Rate Per Distance Mode

5.2.3.2.3.2. Output Files

File Name	Format	Description
RD_[CountyRep]_[yearID]_[monthID]_[TempRange]_in	MySQL	MOVES MySQL Database RPD Table (*_in)
RP_[CountyRep]_[yearID]_[monthID]_[Met-ProfID]_in	MySQL	MOVES MySQL Database RPP SMOKE format Table (*_in)
RV_[CountyRep]_[yearID]_[monthID]_[Temp-Bin]_in	MySQL	MOVES MySQL Database RPV Table(*_in)

5.2.3.3. MOVES Runspec batch script

5.2.3.3.1. Description

The MOVES Runspec batch script file "runspec.csh" is used to execute the MOVES runs for the reference county and reference fuel month.

The output directories are named according to the following naming convention: [CountyRep]_[monthID]_[YearID]_[PollutantID], ie. 13121_Apr_2009_ozone

The output files include:

• rateperdistance.[frm|MYD|MYI]

- ratepervehicle.[frm|MYD|MYI]

- rateperprofile.[frm|MYD|MYI]

5.2.3.3.2. Processing Order

runspec.csh is created by **Runspec_generator.pl** and should be run after importer.csh and before **moves2smkEF.pl**. The runspec.csh is written in Linux c-shell script format depending on which platform is being used (this option is specified by the user in the Runspec_generator.pl) the script is used to run MOVES for the appropriate temperature bins, source types, and poluutants.

The runspec.csh uses following steps:

1. The runspec.csh contains all of the user selections needed to run MOVES using the imported MySQL MOVES databases that were created by importer.csh

2. Creates the MOVES output in ASCII format and also provides the associated XML files for the three different emissions processes: RPD, RPV, RPP.

3. Review log file of the MOVES, called runlog_*.txt.

5.2.3.3.3. Files, Environment Variables, and Run Syntax

5.2.3.3.3.1. Input Files

File Name	Format	Description
runspec.csh	ASCII	runspec script to use the MySQL MOVES databases created by importer.csh and run MOVES
RV_[SCC]_[YR]_[M]_[Temp-Bin]_mrs.xml	XML	importer file specifies the zonemonthhour (*zmh.csv files that will be imported to MOVES for the Rate Per Vehicle Mode
RP_[SCC]_[YR]_[M]_[ProfMY-EAR???]_mrs.xml	XML	importer file specifies the *.csv files that will be imported to MOVES for the Rate Per Process Mode
RD_[SCC]_[YR]_[M]_[TempBin]_[85 or 90?]_mrs.xml	XML	importer file specifies the *.csv files that will be imported to MOVES for Rate Per Distance Mode

5.2.3.3.3.2. Output Files

Note: these output files are placed in separate directories for each reference county, fuel month and pollutant. Example: .../outputs/mysql/13121_Jan_2009_ozone

File Name	Format	Description
rateperdistance.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RPD Table (*.frm,*.MYD,*.MYI)
ratepervehicle.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RPV Table(*.frm,*.MYD,*.MYI)
ratepervehicle_smoke.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RPV SMOKE format Table(*.frm,*.MYD,*.MYI)
rateperprofile.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RPP Table (*.frm,*.MYD,*.MYI)

File Name	Format	Description
rateperprofile_smoke.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RPP SMOKE format Table (*.frm,*.MYD,*.MYI)
activitytype.[frm|MYD|MYI]	MySQL	MOVES MySQL Database activity type Table (*.frm,*.MYD,*.MYI)
agefractotals.[frm|MYD|MYI]	MySQL	MOVES age fraction totals Table(*.frm,*.MYD,*.MYI)
movesactivityoutput.[frm|MYD|MYI]	MySQL	MOVES MySQL Database activity output Table (*.frm,*.MYD,*.MYI)
moveserror.[frm|MYD|MYI]	MySQL	MOVES MySQL Database error Table (*.frm,*.MYD,*.MYI)
moveseventlog.[frm|MYD|MYI]	MySQL	MOVES MySQL Database event log Table(*.frm,*.MYD,*.MYI)
movesoutput.[frm|MYD|MYI]	MySQL	MOVES MySQL Database output Table (*.frm,*.MYD,*.MYI)
movesrun.[frm|MYD|MYI]	MySQL	MOVES MySQL Database run Table (*.frm,*.MYD,*.MYI)
movestablesused.[frm|MYD|MYI]	MySQL	MOVES MySQL Database tables used Table (*.frm,*.MYD,*.MYI)
movesworkersused.[frm|MYD|MYI]	MySQL	MOVES MySQL Database workers used Table(*.frm,*.MYD,*.MYI)
moves.[frm|MYD|MYI]	MySQL	MOVES MySQL Database output Table (*.frm,*.MYD,*.MYI)
rdprex.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RDPREX Table(*.frm,*.MYD,*.MYI)
rpprex.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RPPREX output Table (*.frm,*.MYD,*.MYI)
rvprex.[frm|MYD|MYI]	MySQL	MOVES MySQL Database RVPREX Table(*.frm,*.MYD,*.MYI)
sccvtypeagedistribution.[frm|MYD|MYI]	MySQL	MOVES MySQL Database SCCV Type Age Distribution output Table (*.frm,*.MYD,*.MYI)
sccvtypeagedistribution_final.[frm|MYD|MYI]	MySQL	MOVES MySQL Database SCCV Type Age Distribution Final output Table (*.frm,*.MYD,*.MYI)
tmprdscc.[frm|MYD|MYI]	MySQL	MOVES MySQL Database temporary RD SCC output Table (*.frm,*.MYD,*.MYI)
tmprpscc.[frm|MYD|MYI]	MySQL	MOVES MySQL Database temporary RP SCC output Table (*.frm,*.MYD,*.MYI)
tmprvscc.[frm|MYD|MYI]	MySQL	MOVES MySQL Database temporary RV SCC output Table (*.frm,*.MYD,*.MYI)
tmp_sccroadtypedistribution.[frm|MYD|MYI]	MySQL	MOVES MySQL Database temporary scc road type distribution output Table (*.frm,*.MYD,*.MYI)

File Name	Format	Description
tmp_srcfuelagedistribution.[frm|MYD|MYI]	MySQL	MOVES MySQL Database temporary source fuel age distribution output Table (*.frm,*.MYD,*.MYI)
travelfrac_denominator.[frm|MYD|MYI]	MySQL	MOVES MySQL Database travel fraction denominator Table (*.frm,*.MYD,*.MYI)
travelfrac_numerator.[frm|MYD|MYI]	MySQL	MOVES MySQL Database travel fraction numerator Table (*.frm,*.MYD,*.MYI)
travelfractions.[frm|MYD|MYI]	MySQL	MOVES MySQL Database travel fractions Table (*.frm,*.MYD,*.MYI)

5.2.4. MOVES Post-processing scripts

5.2.4.1. Overview

The MOVES Post-processing scripts consist of two Perl scripts. One is called **"Moves2smkEF.pl"** which extracts the emissions factor tables from the MOVES databases and converts them into a format that can be used as SMOKE inputs. The second script, called **"gen_8digit_scc.pl"**, produces a file mapping 8-digit SCCs to 10-digit SCCs used when importing activity data in **Smkinven**.

5.2.4.2. Moves2smkEF.pl

5.2.4.2.1. Description

The MOVES post-processor script **"Moves2smkEF.pl"** extracts the emission factor tables from the MOVES databases (RPD, RPV, RPP, and RPH MOVES formatted Lookup tables) and converts them into a format that can be processed using **Movesmrg**.

1. Create a new field for 'countyID'

    None of the MOVES lookup tables contain a countyID field solely dedicated to a state-county FIPS code, but SMOKE needs this information. For this reason, **Moves2smkEF.pl** creates a county field by extracting it from other fields that include county codes as part of their value. The RPD table has a linkID field that contains countyID in the leading five digits followed by more numbers to allow for future implementation of link modeling. The RPV table has a zoneID field that is countyID with a trailing zero. The RPP table has a TemperatureProfileID field, which also leads with countyID and is trailed by other numbers to indicate month and other identification. The RPH table has a linkID field the same as the RPD table. The postprocessing script parses countyID from linkID, zoneID and TemperatureProfileID and stores it as a separate field.

2. Map MOVES pollutantID to output pollutant/species name

    The pollutant mapping file has columns for MOVES2014 Pollutant ID, MOVES2014 Pollutant Name, and SMOKE Pollutant/Model Species Name. **Moves2smkEF.pl** reads this list to determine which MOVES pollutants to output in the emission factor files and what name to use for each pollutant. Because MOVES2014 contains emission factors for both inventory pollutants and model species, the sample pollutant mapping file included with SMOKE uses the _INV suffix to distinguish inventory pollutants from model species.

3. Cross-tab pollutantID to reduce output table size

Each of the four MOVES lookup tables contains a pollutantID field, and associated emission rates for each are listed in the ratePerDistance, ratePerVehicle, or ratePerHour field (depending on lookup table). This table structure repeats the leftmost fields (e.g., MOVESRunID, yearID, monthID) for each pollutantID, which greatly increases the total number of rows. The cross-tabbing postprocessing step replaces the column for 'ratePerDistance' (or 'ratePerVehicle' or 'ratePerHour') with additional pollutant-specific columns containing values of emission rates. For example, new field headings would include 'CO_INV', 'NOX', 'BENZENE_INV', etc. Performing this cross-tab on the pollutant emission rates significantly reduces the size of the tables for use by SMOKE.

4. Aggregate emission factors by fuel type, source type, road type, and/or process type

To reduce the total number of sources to be processed in SMOKE, **Moves2smkEF.pl** supports user-specified aggregations of MOVES fuel types, source types, road types, and process types to SCCs.

5. Apply formulas to calculate additional inventory pollutants or model species

Formulas can be applied to augment the existing MOVES output pollutants. For example, speciated versions of criteria pollutants are not provided in the MOVES2014 output nor are speciated emission factors for brake and tire wear.

6. Create final emission rate lookup tables in SMOKE format

The four postprocessed tables named rateperdistance_smoke, ratepervehicle_smoke, rateperprofile_smoke, and rateperhour_smoke are exported from MySQL into ASCII files for their use in SMOKE.

5.2.4.2.2. Processing Order

Moves2smkEF.pl is run after **Runspec_generator.pl**, and before any SMOKE processing. The **Moves2smkEF.pl** is written in Perl, generates SMOKE-ready MOVES lookup tables (i.e., RPD, RPV, RPP, and RPH) by each reference county and fuel month for inputs to **Movesmrg**.

5.2.4.2.3. Files, Environment Variables and Run Syntax

5.2.4.2.3.1. Input Files

File Name	Format	Description
rateperdistance ratepervehicle rateperprofile rateperhour	MOVES format	Columns in the MOVES Emission Rate Lookup Table are described in Table 2.14, "MOVES Emission Rate Lookup Table (MOVES Format)" [89].
InputDBList:	ASCII	List of MySQL database names to process generated by **Runspec_generator.pl**
PollutantFormulasFile:	ASCII	List of formulas used to calculate additional emission factors
FuelTypeMappingFile:	ASCII	List of MOVES fuel type IDs and corresponding aggregated fuel type ID
SourceTypeMappingFile:	ASCII	List of MOVES source type IDs and corresponding aggregated source type ID
RoadTypeMappingFile:	ASCII	List of MOVES road type IDs and corresponding aggregated road type ID

File Name	Format	Description
`Pro-cessTypeMap-pingFile:`	ASCII	List of MOVES process type IDs and corresponding aggregated process type ID
`PollutantMap-pingFile:`	ASCII	List of MOVES pollutant IDs and corresponding pollutant name; only listed pollutants will be included in output emission factor tables

5.2.4.2.3.2. Run Command syntax

```
moves2smkEF.pl [-r RPD|RPV|RPP|RPH] [--formulas PollutantFormulasFile] [--fuel_agg Fu
```

- RPD|RPP|RPV|RPH - optional type of emission factors to process (rate-per-distance, rate-per-vehicle, rate-per-profile, or rate-per-hour); if not specified, script will process all four types

- PollutantFormulasFile - list of formulas used to calculate additional emission factors

- FuelTypeMappingFile - list of MOVES fuel type IDs and corresponding aggregated fuel type ID

- SourceTypeMappingFile - list of MOVES source type IDs and corresponding aggregated source type ID

- RoadTypeMappingFile - list of MOVES road type IDs and corresponding aggregated road type ID

- ProcessTypeMappingFile - list of MOVES process type IDs and corresponding aggregated process type ID

- InputDBList - list of MySQL database names to process (generated by runspec_generator.pl MOVES preprocessor)

- PollutantMappingFile - list of MOVES pollutant IDs and corresponding pollutant name

- OutputPath - optional; overrides the output path specified in InputDBList

5.2.4.2.3.3. Output Files

File Name	Format	Description
`rateper-distance_smoke*.csv`	ASCII	RPD SMOKE formatted lookup table (fuel month specific). The format of MOVES lookup tables are described in Table 2.15, "MOVES Emission Rate Lookup Table (SMOKE Format)" [90].
`rateperve-hicle_smoke*.csv`	ASCII	RPV SMOKE formatted lookup table (fuel month specific)
`rateperpro-file_smoke*.csv`	ASCII	RPP SMOKE formatted lookup table (fuel month specific)
`rateper-hour_smoke*.csv`	ASCII	RPH SMOKE formatted lookup table (fuel month specific)
`mcrlist.rate-per-distance*.lst:`	ASCII	list of MOVES-based rateperdistance lookup tables

File Name	Format	Description
`mcrlist.rate-perve-hicle*.lst:`	ASCII	list of MOVES-based ratepervehicle lookup tables
`mcrlist.rate-perpro-file*.lst:`	ASCII	list of MOVES-based rateperprofile lookup tables
`mcrlist.rate-perhour*.lst:`	ASCII	list of MOVES-based rateperhour lookup tables

5.2.4.3. gen_8digit_scc.pl

5.2.4.3.1. Description

The **gen_8digit_scc.pl** script generates a file mapping 8-digits SCCs to aggregated SCCs produced by **Moves2smkEF.pl**.

5.2.4.3.2. Processing Order

The Perl script performs the following steps:

- Connect to the first database listed in the InputDBList file

- Query the specified database table (rateperdistance, ratepervehicle, rateperprofile, or rateperhour) and generate a list of all SCCs in the MOVES2014 output

- Apply the fuel type, source type, road type, and process type mappings to generate a list of 10-digit aggregated SCCs

- Output the list of 10-digit aggregated SCCs with corresponding 8-digit SCC (first 8 digits of 10-digit SCC)

5.2.4.3.3. Files, Environment Variables and Run Syntax

5.2.4.3.3. Files, Environment Variables and Run Syntax

File Name	Format	Description
`FuelTypeMap-pingFile:`	ASCII	List of MOVES fuel type IDs and corresponding aggregated fuel type ID
`SourceTypeMap-pingFile:`	ASCII	List of MOVES source type IDs and corresponding aggregated source type ID
`RoadTypeMap-pingFile:`	ASCII	List of MOVES road type IDs and corresponding aggregated road type ID
`Pro-cessTypeMap-pingFile:`	ASCII	List of MOVES process type IDs and corresponding aggregated process type ID
`InputDBList:`	ASCII	List of MySQL database names to process generated by **Runspec_generator.pl**; only the first database listed will be used

5.2.4.3.4. Run Command syntax

```
gen_8digit_scc.pl -r RPD|RPV|RPP|RPH [--fuel_agg FuelTypeMappingFile] [--src_agg Sour
```

- RPD|RPP|RPV|RPH - emission factor table to query (rateperdistance, ratepervehicle, rateperprofile, or rateperhour)

- FuelTypeMappingFile - list of MOVES fuel type IDs and corresponding aggregated fuel type ID

- SourceTypeMappingFile - list of MOVES source type IDs and corresponding aggregated source type ID

- RoadTypeMappingFile - list of MOVES road type IDs and corresponding aggregated road type ID

- ProcessTypeMappingFile - list of MOVES process type IDs and corresponding aggregated process type ID

- InputDBList - list of MySQL database names to process (generated by runspec_generator.pl MOVES preprocessor); only first database listed will be used

5.2.4.3.5. Output Files

gen_8digit_scc.pl outputs a list of 10-digit SCCs and corresponding 8-digit SCCs. Users should direct this output to the file of their choosing.

5.3. SMOKE Utility Programs

5.3.1. Overview

There are thirteen programs covered in this section:

Aggwndw is a utility program that can aggregate or window emissions from an I/O API file to a new grid.

BlueSky2Inv converts output files from the BlueSky fire modeling framework into inventory files that SMOKE can read.

The **CEMScan** program calculates summed annual NOx emissions, SO_2 emissions, heat input, gross load, and steam load from a year's worth of CEM data. The output from **CEMScan** is used by **Smkinven** to calculate hourly emissions from annual inventory data when processing CEM data.

Gentpro allows the user to use county average meteorology data to estimate temporal profiles of emissions.

Geofac adjusts SMOKE hourly, gridded, speciated emissions in based on a specified geographic region

Invsplit splits an ORL inventory into separate inventories by state, grouping the states as indicated by your inputs.

Layalloc vertically redistributes hourly, gridded, speciated pre-gridded emissions from emission sources that could emit above the surface layer (e.g. Energy and Industrial sources) based on an user-defined layer fraction input.

Metcombine combines 2-D gridded meteorology data with the 1st layer of 3-D gridded meteorology data to create input files for **Met4moves**.

Metscan reads up to one year's worth of first-layer I/O API meteorology data (whether 2-D or 3-D files) and determine the first and last freeze dates of the year, for use by **Tmpbeis3** in setting winter/summer emission factors by day and grid cell.

Pktreduc reduces the size of /PROJECTION/ packets, but keeps the same information by identifying and implementing state defaults where none already exist in the file.

Smk2emis converts SMOKE netCDF files into a UAM-, REMSAD-, or CAM$_X$-ready emissions file.

Surgtool inputs SMOKE-formatted gridding surrogates for a "fine" input grid and outputs a surrogate file for a "coarse" output grid. It produces approximate "coarse" grid surrogates, the accuracy of which depends on how fine the resolution of the input grid is relative to that of the output grid.

Uam2ncf converts UAM-ready emissions files into I/O API format.

5.3.2. Aggwndw

5.3.2.1. Description

Aggwndw aggregates and/or windows emissions from an I/O API file to a new grid. The program does not change grid projections or interpolate data; the grid boundaries of the input and output grids must be aligned and the output grid must be entirely contained within the input grid. **Aggwndw** automatically checks that the input and output emissions do not differ by more than 0.01% and also produces a report file containing summed input and output data.

5.3.2.2. Processing Order

Aggwndw can be run on any gridded I/O API file. Usually it would be used to create coarse-grid or windowed emissions from fine-grid model-ready output files created by **Smkmerge** or **Mrggrid**.

5.3.2.3. Files and Environment Variables

5.3.2.3.1. Input Files

File Name	Format	Description
GRIDDESC	ASCII	Grid description file containing information about the output grid
INFILE	I/O API NetCDF	Input gridded emissions file

5.3.2.3.2. Input Environment Variables

- IOAPI_GRIDNAME_1: [default: none]

  Sets the name of the output grid whose parameters will be read from the GRIDDESC file.

5.3.2.3.3. Output Files

File Name	Format	Description
OUTFILE	I/O API NetCDF	Gridded emissions file
REPFILE	ASCII	Report file; provides summed input and output emissions for QA purposes

5.3.3. BlueSky2Inv

5.3.3.1. Description

BlueSky2Inv converts output files from the BlueSky fire modeling framework into an annual IDA-formatted inventory and an hourly emissions inventory.

5.3.3.2. Processing Order

BlueSky2Inv is used to create inventories from BlueSky output for input into **Smkinven**.

5.3.3.3. Files and Environment Variables

5.3.3.3.1. Input Files

File Name	Format	Description
COSTCY	ASCII	Country, state, and county codes file; used to assign time zones to each source based on the source's state and county codes
FILELIST	ASCII	List of BlueSky output files including path and filename

5.3.3.3.2. Input Environment Variables

- G_STDATE: [default: none]

  Starting date of data (YYYYDDD); used to set the inventory year in the annual inventory. G_STDATE is also used to create default entries in the daily inventory file for each source.

- VARLIST: [default: none]

  Selects the variables that will be output in the annual and daily inventories. Valid values are AREA, HFLUX, PM2_5, PM10, PM, PMC, CO, CO2, CH4, NMHC, and TOG. The values for the variable should be formatted as follows:

  setenv VARLIST "AREA, HFLUX, PM2_5, PMC, CO, TOG"

5.3.3.3.3. Output Files

File Name	Format	Description
PTDAY	ASCII	Daily emissions inventory
PTINV	ASCII	Annual IDA-formatted inventory

5.3.4. CEMScan

5.3.4.1. Description

CEMScan reads a year's worth of CEM data and calculates summed annual NOx emissions, SO_2 emissions, heat input, gross load, and steam load for each uniquie ORIS/boiler combination in the output CEM data. While writing the output, CEMScan will skip any ORIS/boiler combinations where the total annual NOx emissions are zero or null or where the total annual gross load, steam load, and heat input are zero or null. The program will write these ORIS/boiler combinations to the report file along with a flag indicating that

the combination was not written to the output file. The output from **CEMScan** is used by **Smkinven** to calculate hourly emissions from annual inventory data when reading CEM data. Detail information is described in Section 2.9.19, "Processing hour-specific CEM data" [62].

5.3.4.2. Processing Order

CEMScan should be run before **Smkinven** when processing CEM data.

5.3.4.3. Files and Environment Variables

5.3.4.3.1. Input Files

File Name	Format	Description
FILELIST	ASCII	List of CEM input files including path and filename; the files must contain data for an entire year to be consistent with the annual inventory

5.3.4.3.2. Input Environment Variables

- MAX_CEM_UNITS: [default: none]

    Maximum number of CEM units in input data; this value can usually be based on the number of sources in the annual inventory

5.3.4.3.3. Output Files

File Name	Format	Description
OUTFILE	ASCII	CEM summary file
REPFILE	ASCII	Report summarizes CEM data

5.3.5. Gentpro

5.3.5.1. Description

The **Gentpro** program estimates emissions temporal profiles for residential wood combustion (RWC), agricultural ammonia (RC_NH3), livestock ammonia (BASH_NH3) or generic area sources by relating meteorology to air emission fluxes. **Gentpro** reads in hourly gridded meteorology data from the Meteorology-Chemistry Interface Processor (MCIP) and a gridded spatial surrogate (commonly used in SMOKE) to produce temporal profiles and cross-reference data in a comma separated value (CSV) format and/or a temporal data binary netCDF file. Annual MCIP input data are required to calculate temporal profiles with **Gentpro**. Spatial surrogate(s) are used to identify the grid cells for computing county averaged meteorology variables.

Gentpro provides an option to specify a profile method [PROFILE_METHOD], which can be set to generate RWC, BASH_NH3, RC_HNH3, or generic meteorology-based temporal profiles. the RWC, BASH_NH3 and RC_NH3 temporal profiles are based on algorithms described by Adelman (2009 and 2010), Mansell et al. (2000), and Bash et al. (2012) respectively, while the MET profiles can ben generated from any user-selected meteorology variable. The RWC profile method produces county-specific daily temporal profiles to convert annual or monthly inventories to daily emissions estimates. Diurnal profiles must then be applied to estimate hourly RWC emissions from the daily estimates. **Gentpro** provides the option to aggregate the daily RWC profiles to month-of-year profiles as well, which can be used to compute monthly inventory

estimates from annual inventory data. The BASH_NH3 and RC_NH3 profile methods produce county-specific hourly temporal profiles to convert annual, monthly, or daily inventories to hourly emissions estimates. **Gentpro** provides the option to aggregate the hourly profiles to day-of-year or month-of-year profiles as well. The MET profile method represents time series of a selected meteorology variable as a temporal profile and can be used to convert annual, monthly, or daily inventories to hourly emissions estimates. **Gentpro** also provides the option to aggregate the hourly MET profiles to day-of-year or month-of-year profiles. All of the **Gentpro** profile options create temporal cross-reference files that associate county FIPS codes in a modeling domain to the new profiles.

[1] RWC

A regression equation relating observed daily minimum temperatures and chemical tracers of wood combustion was developed using data from sites in the Pacific Northwest of the United States (Adelman, 2009; Adelman, 2010). The sites selected to develop the algorithm were screened to be representative of RWC activities. Equation (1) uses estimates of daily PM2.5 emissions to build either annual-to-daily or annual-to-monthly emission profiles for application to RWC sources. The algorithm uses county average daily or monthly minimum temperatures to estimate the percentage of RWC emissions allocated to each day or month of the year. **Gentpro** includes a temperature threshold setting to limit RWC emissions from allocated to days too warm to have much (if any) RWC emissions. A temperature threshold (T_t) variable is implemented in **Gentpro** to allow users to set a maximum temperature cutoff by either state or county. By default, **Gentpro** estimates RWC emissions at temperatures below 50°F. If the temperature threshold variable is set in the Gentpro script, RWC emissions will be activated below T_t. A look-up table of state/county FIPs codes and temperatures (°F) can be input to **Gentpro** to set several different temperature thresholds across a modeling domain. The purpose of using the temperature threshold override is to control when RWC emissions are activated in regions where the 50°F cutoff may not be appropriate. For example, counties in southern Florida may only experience one or two days in an entire year when the minimum temperature dips below 50°F. Without using the threshold override, all of the RWC emissions for these counties will be allocated to these one or two days, producing large emissions spikes that may not be realistic. By increasing the temperature threshold, these spikes can be avoided by activating the RWC algorithm on other days. The outputs from this algorithm are temporal profiles that convert annual or monthly RWC inventories to daily emissions for every county in the modeling domain.

Two alternative RWC temporal profile algorithms are implemented in **Gentpro**. Equation 1 below shows the algorithm developed from the Pacific Northwest Sites described by Adelman (2009; 2010).

The second RWC temporal algorithm is an ad hoc approximation of the original equation that broadens its application by removing the intercept term. By removing the intercept term, Equation 2 below will not produce negative emissions when the temperature threshold is increased above 53.3°F.

The alternate RWC algorithms can be selected using `RWC_ALT_EQ_YN` listed below. When `RWC_ALT_EQ_YN` = N, Equation 1 will be used to calculate RWC temporal profiles.

1. **Original RWC Algorithm developed by Adelman (2009;2010)**

    When $T_{i,d}$ is equal or less than 50°F,

    $PE_{i,d}$ = (42.12 - 0.79 x $T_{i,d}$) / Sum (42.12 - 0.79 x $T_{i,d}$)

    When $T_{i,d}$ is greater than T_t,

    $PE_{i,d}$ = 0.0

    When $T_{i,d}$ is greater than 50°F and $T_{i,d}$ is less than T_t,

    $PE_{i,d}$ = (42.12 - 0.79 x 50) / Sum (42.12 - 0.79 x 50)

2. **Alternative RWC Algorithm**

When $T_{i,d}$ is equal or greater than T_t,

$PE_{i,d} = 0.0$

When $T_{i,d}$ is less than T_t,

$PE_{i,d} = [0.79 \times (T_{i,t} - T_{i,d})] / Sum [0.79 \times (T_{i,t} - T_{i,d})]$

where

- $PE_{i,d}$ = Percentage of emissions in county i on day d

- $T_{i,d}$ = Daily minimum ambient temperature (°F) in county i on day d

- $T_{i,t}$ = Daily threshold ambient temperature (°F) in county i

- T_t = Daily threshold ambient temperature (default = 50°F)

- Note: $T_{i,m}$ can be replaced by monthly minimum ambient temperature in county i on month m

[2] NH3

Two NH3 profile methods are implemented in **Gentpro** to estimate hourly temporal allocation factors, which are output as temporal profiles to convert annual, monthly, or daily livestock ammonia inventories to hourly emissions for every county in the modeling domain. **Gentpro** can output hour-of-month and hour-of-year temporal profiles that can be input to the **Temporal**. The recommended application of the BASH_NH3 and RC_NH3 profile methods are to convert monthly agricultural ammonia inventories to hourly emissions.

1. RC_NH3 profile method: Russell and Cass (1986) developed a theoretical equation based on investigations of NH3 emissions from animal waste decomposition conducted by Muck and Steenhuis (1982) to predict diurnal NH3 emission variations as a function of daily meteorology. The Russell and Cass equation (a.k.a RC_METHOD) listed below is adapted for this application to use county average meteorology and implemented in **Gentpro** to estimate hourly emissions from county annual, monthly, or daily agricultural livestock NH_3 inventories.

   $E_{i,h} = [2.36^{(T_{i,h}-273)/10}] \times V_{i,h}$

2. BASH_NH3 profile method: Zhu, Henze, et al. (2013) developed an empirical equation based on observations from the TES instrument with the GEOS-Chem model and its adjoint to estimate diurnal NH3 emission variations from livestocks as a function of ambient temperature and aerodynamic resistance meteorological variable. The equation is listed below.

   $E_{i,h} = [161500/T_{i,h} \times e^{(-1380/T_{i,h})}] \times AR_{i,h}$

- $PE_{i,h} = E_{i,h} / Sum(E_{i,h})$

where

- $PE_{i,h}$ = Percentage of emissions in county i on hour h

- $E_{i,h}$ = Emission rate in county i on hour h

- $T_{i,h}$ = Ambient temperature (Kelvin) in county i on hour h

- $V_{i,h}$ = Wind speed (meter/sec) in county i (minimum wind speed is 0.1 meter/sec)

- $AR_{i,h}$ = Aerodynamic resistance in county i

[3] MET: Generic Meteorology Profile Algorithm

This is a simple algorithm that computes the percentage of emissions to allocate to each hour based on the hourly fractional contribution of a selected meteorology variable relative to the annual, monthly, and daily sum of all hourly values for that variable. While the fractional contribution of county-average wind speed for each hour of the year is somewhat meaningless, it does provide a way to build a time series that can be applied to an annual/monthly inventory to estimate hourly values using monthly, daily and/or hourly MET profiles.

1. $PE_{i,h} = m_{i,h} / Sum(m_{i,h})$

where

- $PE_{i,h}$ = Percentage of emissions in county i on hour h

- $m_{i,h}$ = Meteorology variable in county i on hour h

[Reference]

Adelman, Z. (2010) 'Proposal: Estimating Temporal Profiles for Residential Wood Combustion Emissions', Prepared for U.S. EPA, OAQPS, Research Triangle Park, NC under Contract No. EP-D-07-102 Assignment No. 3-07 (M. Houyoux, WAM) by UNC, Chapel Hill, NC.

Adelman, Z. (2009) 'Technical Memo: Emissions Modeling Improvements Task 2 – Temporal Allocation for Residential Wood Combustion', Prepared for U.S. EPA, OAQPS, Research Triangle Park, NC under Contract No. EP-D-07-102 Assignment No. 2-13 (M. Houyoux, WAM) by UNC, Chapel Hill, NC.

Mansell, G, S. Kemball-Cook, L. Ma (2009) 'Draft Report: Temporal Allocation of Agricultural Ammonia Emissions', Prepared for U.S. EPA, OAQPS, Research Triangle Park, NC under Contract No. EP-D-07-102 Assignment No. 2-13 (M. Houyoux, WAM) by ENVIRON, Novato, CA.

Muck, R.E., and T.S. Steenhuis (1982) 'Nitrogen Losses from Manure', Agricultural Wastes, 4: 41-54.

Russell, A.G. and G.R. Cass (1986) 'Verification of a Mathematical Model for Aerosol Nitrate and Nitric Acid Formation and Its Use for Control Measure Evaluation', Atmospheric Environment, 20: 2011-2025.

Zue, Henze, et al (2013) 'Constraining U.S. Ammonia Emissions using TES Remote Sensing Observations and the GEOS-Chem adjoint model', Journal of Geophysical Research: Atmospheres, 118: 1-14.

5.3.5.2. Processing Order

This optional program must run prior to **Temporal**.

5.3.5.3. Files and Environment Variables

Figure 5.1. Gentpro input and output files
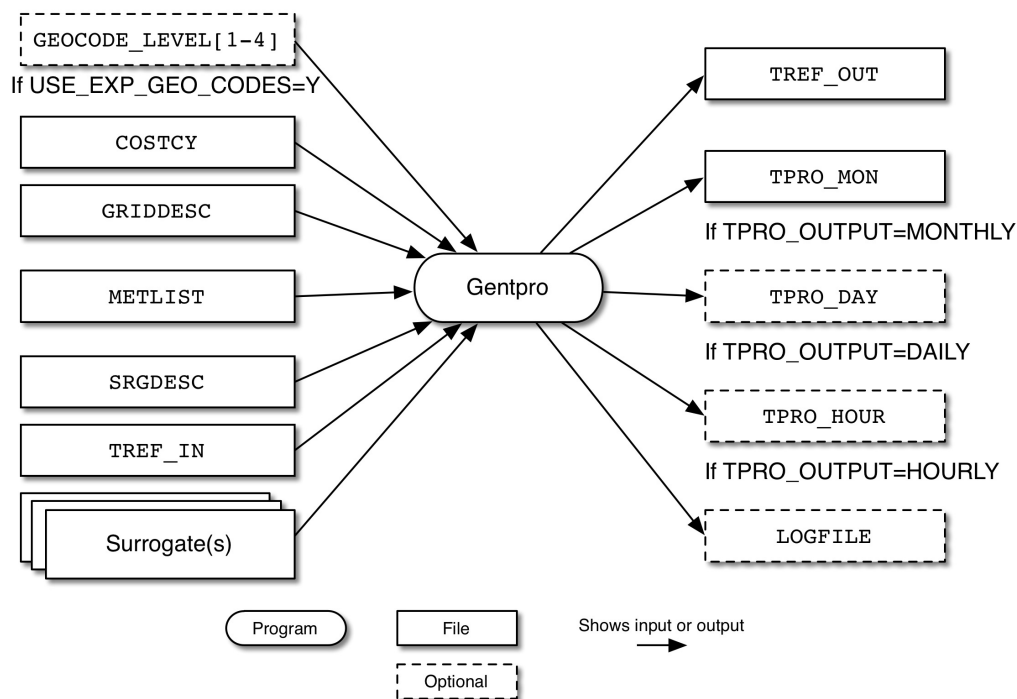
5.3.5.3.1. Input Files

File Name	Format	Description
COSTCY	ASCII	Five digit code for country, state and county
GEO-CODE_LEVEL[1-4]	ASCII	Geographical level codes and descriptions for gridded inventories, only when USE_EXP_GEO_CODES is set to Y, (COSTCY is ignored when Section 8.10.3, "GEOCODE_LEVEL[1-4]: Geographical Input Files" [447] files are used)
GRIDDESC	ASCII	Grid descriptions file that defines the output grid for the output surrogates file created by the program. It must include the name of the grid set by the IOAPI_GRIDNAME _1 environment variable setting.
METLIST	ASCII	A list of meteorology files containing temperature, mixing ratio, and surface pressure data created by **Metcombine**
RWC_COUNTY_TEMP (optional)	ASCII	County-specific minimum temperature for RWC equations (see Section 8.10.9, "RWC_COUNTY_TEMP: County-specific minimum temperature for Residential Wood Combustion (RWC) Equations" [456])
SRGDESC	ASCII	Spatial surrogate description file produced by MCIP (see Section 8.4.1, "SRGDESC: Surrogate Designation, Description and Surrogate profile filename." [387])

File Name	Format	Description
Surrogate File(s)	ASCII	Spatial surrogate file(s) selected by `SRG_LIST`
`TREF_IN`	ASCII	Temporal cross-reference input file (see Section 8.3.2, "`[A|M|P]TREF`: Temporal cross-reference file" [383])

5.3.5.3.2. Input Environment Variables

- `AERO_RESISTANCE_VAR`: [default: blank]

  Define the aerodynamic resistance variable name for Bash's algorithm for livestock ammonia sources

- `SLOPE`: [default: 0.79]

  Define the value of the slope of RWC equation

- `CONSTANT`: [default: 42.12]

  Define the value of constant of RWC equation

- `ENDDATE`: [default: 0]

  Defines the last day of year (December 31st) for a proper temporal profile calculation. MCIP file must cover last hour of December 31st of all counties in modeling output time zone (i.e., GMT).

- `PROFILE_METHOD`: [default: MET]

  Indicates which of the three temporal profile types to generate by **Gentpro**

  - RWC: use the Adelman's regression equation to calculate temporal profiles for residential wood combustion (RWC) sources

  - BASH_NH3: use the Bash's algorithm to calculate temporal profiles for livestock ammonia sources

  - RC_NH3: use the Russell and Cass algorithm to calculate temporal hourly profiles for agricultural ammonia sources

  - MET: use the time series of the selected meteorology variable to calculate temporal profiles

- `RWC_COUNTY_TEMP_YN`: [default: N]

  Allows the user to determine county-specific minimum temperature for RWC estimate equations. It will override default minimum temperature [`DEFAULT_TEMP_RWC`] listed above.

  - Y: use the county-specific minimum temperature from `RWC_COUNTY_TEMP` input file.

  - N: do not use the county-specific minimum temperature for RWC equations.

- `RWC_EQ_ALT_YN`: [default: Y]

  Allows the user to choose the alternative RWC equation listed above to avoid negative RWC emissions.

  - Y: use the alternative RWC equation to compute RWC emissions.

  - N: use the original RWC equations to compute RWC emissions.

- `SCC_LIST`: [default: blank]

Defines a comma-separated list of SCCs that will use the temporal profiles generated by **Gentpro**, example: setenv SCC_LIST "2104008000,2104008001,2104008002"

- `SRG_LIST`: [default: 0]

  Defines a comma-separated list of SCCs that will use the temporal profiles generated by **Gentpro**, example setenv SRG_LIST "340,100"

- `SRGDESC`: [default: none]

  Directory path and name of the spatial surrogate file for computing averaged meteorology.

- `SRGPRO_PATH`: [default: blank]

  Defines the location of spatial surrogate files

- `STDATE`: [default: 0]

  Defines the first day of year (January 1st) for a proper temporal profile calculation. MCIP file must cover first hour of January 1st of all counties in modeling output time zone (i.e., GMT).

- `TEMP_VAR`: [default: blank]

  Must define the name of variable for the all PROFILE_METHOD settings to use in the profile calculations. The program looks for the selected meteorology variables first in the METLIST file (set below) that contains a list of Metcombine program output files or MET files that contain targeted variables from the MET_CRO_3D or/and MET_CRO_2D files. If it doesn't find the variable(s) in either file, it stops with an error.

- `TPRO_OUTPUT`: [default: ALL]

  Specifies the type of temporal profiles to produce

  - MONTHLY: generate month-of-year temporal profiles

  - DAILY: generate both month-of-year and day-of-month temporal profiles

  - HOURLY: generate hour-of-year, hour-of-month, and hour-of-day temporal profiles

    - Not applicable when PROF_METHOD = RWC

  - ALL: generate all possible profiles for the selected profile method

    - No hourly profiles will be generated when PROF_METHOD = RWC

- `TREF_IN`: [default: blank]

  Defines an input temporal cross-reference file for supplying monthly, weekly, and diurnal profile IDs. The TREF_IN file will be scanned for the FIPS/SCC combinations of sources associated with profiles generated by **Gentpro**. If there are no entries in TREF for the specific FIPS/SCC combinations being simulated with **Gentpro**, the program will look for default cross-reference entries (FIPS=0, SCC=0). Examples of how the TREF file will be used by **Gentpro** include:

  - If TPRO_OUTPUT = DAILY, month-of-year and day-of-month temporal profiles will be generated by **Gentpro** and diurnal profiles will be needed to estimate hourly emissions. In this case, the diurnal profile IDs for the applicable FIPS/SCC combinations in TREF_IN will be written to the temporal cross reference file output by **Gentpro**

- If TPRO_OUTPUT = MONTH, month-of-year profiles will be generated by **Gentpro** and both weekly and diurnal profiles will be needed to estimate hourly emissions. In this case, the weekly and diurnal profile IDs for the applicable FIPS/SCC combinations in TREF_IN will be written to the temporal cross-reference file output by **Gentpro**.

- If PROFILE_METHOD = BASH_NH3 or RC_NH3, hour-of-year profiles will be generated by **Gentpro** to estimate hourly emissions using monthly inventory

- If PROFILE_METHOD = MET and TPRO_OUTPUT = HOURLY[or ALL], hour-of-year, hour-of-month and hour-of-day temporal profiles will be output to netCDF TPRO_HOUR by **Gentpro**

- USE_EXP_GEO_CODES: [default: N]

  If set to Y then use GEOCODE_LEVEL[1-4] files instead of the COSTCY file to define expanded geographical detail such as country-level(GEOCODE_LEVEL1), country/state-level (GEO-CODE_LEVEL2), country/state/county-level (GEOCODE_LEVEL3) or country/state/county/tribal-level (GEOCODE_LEVEL4).

- WSPEED_VAR: [default: blank]

  Defines the name of wind speed variable from MET files. It is only used when PROFILE_METHOD is set to "RC_NH3". example: setenv WSPEED_VAR "WSPD10"

5.3.5.3.3. Output Files

File Name	Format	Description
TPRO_MON	ASCII	Month-of-year temporal profiles
TPRO_DAY	ASCII	Day-of-month temporal profiles
TPRO_HOUR	I/O API netCDF	Hourly temporal profiles
TREF_OUT	ASCII	Temporal cross-references input file.
LOGFILE	ASCII	Log generated from executing **Gentpro**

5.3.6. Geofac

5.3.6.1. Description

The **Geofac** program provides a means for adjusting emissions by grid cell after SMOKE has already created model-ready inputs. For example, the program could be used to zero emissions from all grid cells that intersect a certain state, which would be a rough approximation of removing that state's inventory from the model-ready files. Although it is more accurate to modify the emissions using the **Cntlmat** program, that approach takes longer and is possible only if the SMOKE input files used to create the model ready emissions are still available. **Geofac** is useful in situations where speed and simplicity are more important than the accuracy of the emissions, or when only the gridded model-ready inputs are available and the inventories used to create them are not.

Geofac reads in a gridded 2-D or 3-D SMOKE emissions file (output from the **Smkmerge** program) and multiplies the emissions with a user-supplied adjustment factor for each individual species in the emissions file. The factors are assigned by grid-cell regions, which are defined by the input "mask" file. Each grid cell either receives the factors or it does not (the mask is simply an on-off switch). The resulting hourly, gridded, speciated emissions data are output to a new I/O API file with the same spatial and temporal resolution and the same model species with different values.

5.3.6.2. Processing Order

This optional program can be run after **Smkmerge** and before either **Mrggrid** (if it is being used to merge several **Smkmerge** output files together) or the AQM of interest.

5.3.6.3. Files and Environment Variables

5.3.6.3.1. Input Files

File Name	Format	Description
GEOMASK	I/O API NetCDF	Gridded, time-independent file that assigns a 1 to any grid cell whose emissions you want to adjust. The emissions from IN-FILE in grid cells that have a 1 in GEOMASK are multiplied by the factor provided in SPECFACS. All other emissions and grid cells are untouched.
INFILE	I/O API NetCDF	Hourly, gridded, speciated inputs from **Smkmerge**. INFILE should be set to logical file name AGTS_[L|S], BGTS_[L|S], MGTS_[L|S], PGTS_[L|S], PGTS3D_[L|S], EGTS_[L|S], or EGTS3D_[L|S], which are the output files from **Smkmerge**.
SPECFACS	ASCII	File that lists the species name and the factor to be applied to all grid cells with a 1 value in the GEOMASK file. The species name must be in columns 1-16 and the factor in columns 17-22. For example, the following SPECFACS file sample would multiply NO and NO_2 emissions by 1.3 (30% increase) and multiply PAR by 0.6 (40% decrease). `NO             1.3` `NO2            1.3` `PAR            0.6`

5.3.6.3.2. Input Environment Variables

None

5.3.6.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Geofac**
OUTFILE	I/O API NetCDF	Hourly, gridded, speciated emissions, adjusted based on GEO-MASK and SPECFACS.

5.3.7. Invsplit

5.3.7.1. Description

The **Invsplit** program splits an ORL inventory file into multiple raw inventory files, based on the state codes. The number of output files is set at run time based on the input file that defines the splitting. Up to 80 new files can be created from a single inventory file, which is more than enough files to split a U.S.

inventory into a separate file for each state. The program works on area, on-road mobile, nonroad mobile, and point inventories. Separating a single inventory file into multiple files allows the inventory data to be recombined with other data more easily.

5.3.7.2. Processing Order

This optional program can be run prior to running the **Smkinven** program to import the emission inventories.

5.3.7.3. Files and Environment Variables

5.3.7.3.1. Input Files

File Name	Format	Description
ARINV, MBINV, or PTINV	ASCII	Input inventory file; the logical name used depends on the SMK_SOURCE setting. The ARINV file is for nonpoint (stationary area) and nonroad mobile sources, the MBINV file is for on-road mobile sources, and the PTINV file is for point sources.
SPLITS	ASCII	Space-delimited file that lists the state codes and indicates which output file should include the emissions for that state. The first column is for the state codes and the second column is the output file number. In this example, states 1 and 4 will be written to the OUTFILE01 output file logical name, state 37 to OUTFILE02, and state 45 to OUTFILE03. The numbers in the second column need not be sequential. 01 1 04 1 37 2 45 3

5.3.7.3.2. Input Environment Variables

None

5.3.7.3.3. Output Files

File Name	Format	Description
OUTFILEnn	ASCII	One or more inventory files in the same inventory format input by the ARINV, MBINV, or PTINV input file for a given program run. The actual logical file name of these files is OUTFILE01, OUTFILE02, OUTFILE03, ... OUTFILEnn, where nn = the largest number in the second column of the SPLITS file. The output file names are not necessarily sequential, depending on whether or not sequential numbers are used in the second column of the splits file.
LOGFILE	ASCII	Log generated from executing **Invsplit**

5.3.8. Layalloc

5.3.8.1. Description

The **Layalloc** program calculates 3-D emissions from a 2-D pre-gridded emissions file. **Layalloc** is used to reallocate single layer 2-D surface layer emissions into multiple model layers. This approach is useful for processing pre-gridded or industrial area source inventories that do not contain stack information for plume rise treatment but represent elevated sources.

Layalloc reads in a gridded 2-D pre-gridded emissions file (either output from the **Smkmerge** program or from a global emission inventory (EDGAR)) and redistributes the emissions into multiple model layers based on a user-defined vertical profile. The vertical profiles that are input to **Layalloc** consist of a layer bottom height, layer top height, and the fraction of emissions to allocate to the layer. SMOKE willl interpolate from the layer heights (in meters) to the model layers (in sigma-pressure). This approach assumes that the vertical factors are uniformly applied over all horizontal grid cells. The resulting hourly, gridded, speciated 3-D emissions data are output to a new I/O API file with the same temporal resolution and the same model species as the original 2-D emissions file.

5.3.8.2. Processing Order

This optional program can be run after **Smkmerge** and before either **Mrggrid** (if it is being used to merge several **Smkmerge** output files together) or the AQM of interest.

5.3.8.3. Files and Environment Variables

5.3.8.3.1. Input Files

File Name	Format	Description
INFILE	I/O API NetCDF	2-D hourly, gridded, speciated inputs from **Smkmerge**. INFILE should be set to logical file name AGTS_[L|S], BGTS_[L|S], MGTS_[L|S], PGTS_[L|S], or EGTS_[L|S], which are the output files from **Smkmerge**.
MET_CRO_3D	I/O API NetCDF	Meteorology file contains a variable "ZF" which is full-layer height above ground (m)

LAYER_FRAC-TION	ASCII	File that lists the layer top and bottom heights in meter(s) along with a percentage factor. For example, the following LAYER_FRACTION file sample would distribute 2-D emissions into multiple model layers based upon five user-defined layers.
		
		`No., Bottom Ht(m), Top Ht(m), Fraction`
		`1,          0,         100,        0.60`
		`2,        100,         200,        0.20`
		`3,        200,         400,        0.10`
		`4,        400,         600,        0.07`
		`5,        600,        1000,        0.03`

5.3.8.3.2. Input Environment Variables

None

5.3.8.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Layalloc**
OUTFILE	I/O API NetCDF	3-D hourly, gridded, speciated emissions, adjusted based on LAYER_FRACTION.

5.3.9. Metcombine

5.3.9.1. Description

Metcombine combines 2-D gridded meteorology data with the 1st layer of 3-D gridded meteorology data to create custom files for on-road mobile processing with MOVES. To calculate humidity data for use in MOVES, the program **Met4moves** needs temperature, pressure, and water vapor mixing ratio data in the same meteorology file. Pressure and mixing ratio data are typically found in the MET_CRO_3D file with the variable names PRES and QV, respectively. The MET_CRO_2D file usually contains ground-level temperatures (TEMPG), 1.5 meter temperatures (TEMP1P5), 2 meter temperatures (TEMP2), and 10 meter temperatures (TEMP10). While it is possible to use air temperature data from the MET_CRO_3D file (variable named TA), it is generally recommended to use temperature data from closer to the ground, i.e. ground or 1.5 meter temperatures from MET_CRO_2D. **Metcombine** is used to create custom meteorology files that contain the variables PRES, QV, and TEMPG (or any other temperature variable).

The program takes two I/O API NetCDF gridded meteorology files as input. The files must have identical grid paramaters, start dates and times, time steps, and run lengths. **Metcombine** is generally used to combine a 2-D file with a 3-D file, but the program is not restricted to that combination; it can be used to combine two 2-D files or the 1st layers of two 3-D files.

5.3.9.2. Processing Order

Metcombine can optionally be run before **Met4moves**. If used, **Metcombine** must be run for each individual meteorology file you need to create. For example, if you have five days worth of meteorology data as one-day files, you'll need to run **Metcombine** five times. If the meteorology variables you wish to use are already in the same file, then you do not need to use **Metcombine**. Example: If you decide to use ambient air temperature (TA) from the same METCRO_3D file that also contains PRES and QV, then you do not need to run Metcombine.

5.3.9.3. Files and Environment Variables

5.3.9.3.1. Input Files

File Name	Format	Description
METFILE1	I/O API NetCDF	First gridded meteorology file; can be either a 2-D or 3-D file
METFILE2	I/O API NetCDF	Second gridded meteorology file; can be either a 2-D or 3-D file

5.3.9.3.2. Input Environment Variables

- VARLIST: [default: none]

  Sets the names of the variables that will be read from METFILE1 and METFILE2. Each variable must occur in exactly one of the meteorology files, i.e. the same variable cannot be in both files.

5.3.9.3.3. Output Files

File Name	Format	Description
OUTFILE	I/O API NetCDF	2-D output meteorology file. This file contains the variables requested in VARLIST; for 3-D data, OUTFILE will only contain the 1st layer of data. OUTFILE will have the same grid parameters, start date and time, time step, and run length as METFILE1 and METFILE2.

5.3.10. Metscan

5.3.10.1. Description

Metscan reads in a single (often annual) temperature I/O API meteorology data file (created from either the MET_CRO_2D or MET_CRO_3D files) and determines the first freeze date and last freeze date of a year, by grid cell. It outputs a gridded file with a daily (24-hour) time step that contains a single variable. For each grid cell, that variable has a value of 0 (zero) when the date being modeled indicates that the cell is experiencing winter conditions; otherwise it has a value of 1, indicating summer conditions. The program can be set to run in the Northern Hemisphere or in the Southern Hemisphere (it will not work on global meteorology datasets). In the Northern Hemisphere, the first freeze date in the fall marks the start of winter, and the last freeze date in the spring marks the end of winter; all other days are considered summer days. The seasons are reversed in the Southern Hemisphere, and the program accommodates this accordingly.

The BEIS3 model operates using the assumption of either winter or summer emission factors. There are no spring or fall factors available, which is why this program is limited to the choice of summer or winter. The use of the freeze date to determine summer or winter emission factors is based on EPA's recommendation.

The program reads in the name of the temperature variable as an option, so that the ground temperature, 1.5-meter temperature, or 10-meter temperature can be used from the MET_CRO_2D file, or the layer-1 temperature can be used from the MET_CRO_3D file. See the TMPR_VAR option below for more information on the names of the temperature variables that are acceptable.

5.3.10.2. Processing Order

Metscan can optionally be used before **Tmpbeis3**, which does not require the output from **Metscan** to run. **Metscan** is needed only when users wish to have **Tmpbeis3** automatically switch between winter and summer emission factors by grid cell. This approach is recommended when processing time periods and modeling domains include temperatures below 32 °F either for a substantial number of days relative to the number of episode days or at critical dates in the episode.

5.3.10.3. Files and Environment Variables

5.3.10.3.1. Input Files

File Name	Format	Description
MET_CRO_3D	I/O API NetCDF	Meteorology file that can be a 2-D or 3-D file (though the logical file name is the same in both cases) and needs to include all dates in the modeling period. For example, for an annual SMOKE run this file would need to have all hours of the year. The file can be created by extracting one or more temperature

File Name	Format	Description
		variables (at a minimum, the one you intend to use) out of the `MET_CRO_2D` or `MET_CRO_3D` files output by the MCIP or MCPL programs using the **m3xtract** I/O API utility and appending all days of data into the same file. The **m3xtract** utility documentation is provided with the I/O API, but we provide some information here. The utility expects input logical file name `INFILE` and creates output logical file name `OUTFILE`. It prompts for the start date, start time, duration, and variables that you want to extract from the `INFILE`. If `OUTFILE` already exists, **m3xtract** appends additional data to the file (assuming that the grid is the same and the time period is sequential).

5.3.10.3.2. Input Environment Variables

- `TMPR_VAR`: [default: TA]

  Sets the name of the temperature variable that will be read from the `MET_CRO_3D` input file. Acceptable values for MCIP and MPCL outputs available at the time of this writing are TA (ambient layer-1 temperature from the `MET_CRO_3D` file), TEMP10 (10-meter temperature from the `MET_CRO_2D` file), TGD (ground temperature), TGRND (ground temperature), and TEMP1P5 (1.5-meter temperature). Current guidance from EPA suggests that a 10- to 20-meter temperature is the best choice for the temperature input to BEIS3, to account for the fact that these models do not include a canopy model.

- `N_S_HEMI`: [default: NORTH]

  Sets the program to run for the Northern or Southern Hemisphere. Acceptable values for this setting are:

  - NORTH: Northern Hemisphere. Program starts July 1 and scans data in reverse chronological order to find the last freeze date and in chronological order to find the first freeze date.

  - SOUTH: Southern Hemisphere. Program starts January 1 and scans data in chronological order to find the first freeze date. It also starts December 31 and scans the data in reverse chronological order to find the last freeze date.

5.3.10.3.3. Output Files

File Name	Format	Description
BIOSEASON	I/O API NetCDF	For each grid cell, contains 0 for winter dates and 1 for summer (nonwinter) dates. The file's duration will be as long as that of the input meteorology file you provide.
LOGFILE	ASCII	Log generated from executing **Metscan**

5.3.11. Pktreduc

5.3.11.1. Description

The **Pktreduc** program reduces the size of growth packets (using the /PROJECTION/ packet in the `GCNTL` file), but keeps the same information. This is done by identifying and implementing state defaults where

possible. This program works solely with growth packets that have state/county and SCC entries only; it will not work if the packet contains SIC, MACT, or pollutant codes.

The program identifies the most common growth factor for each state across the many counties for each SCC, and this factor is assigned as the default for the state/SCC combinations. The county/SCC-specific entries are left as is. In this way, the contents of the file are the same as far as SMOKE is concerned, but the file processes through **Cntlmat** much faster because the growth packet is much smaller. It is acceptable if state defaults are already in the file, and **Pktreduc** will leave these untouched. When state defaults are found for a state (i.e., county code is set to 000), the program skips reductions for that state and no changes are made (even if they could be).

5.3.11.2. Processing Order

The **Pktreduc** program is an optional preprocessor for the `GCNTL` input file to **Cntlmat** when processing growth packets.

5.3.11.3. Files and Environment Variables

5.3.11.3.1. Input Files

File Name	Format	Description
`GCNTL`	ASCII	Growth and controls file that includes only the /PROJECTION/ packet. The other packets that are normally acceptable as input to **Cntlmat** (see Section 8.6.1, "`GCNTL`: Files for controls and/or past- and future-year growth" [395]) should not be included in this file when it is an input to **Pktreduc**. In addition, the /PROJECTION/ packet entries can include only state/county/SCC combinations; no combinations that include SIC, MACT, or pollutant codes or other fields should be included.

5.3.11.3.2. Input Environment Variables

None

5.3.11.3.3. Output Files

File Name	Format	Description
`GCNTL_OUT`	ASCII	Reformatted `GCNTL` file that includes only the /PROJECTION/ packet for inventory growth. The reformatting performed is described in Section 5.3.11.1, "Description" [204].
`LOGFILE`	ASCII	Log generated from executing **Pktreduc**

5.3.12. Smk2emis

5.3.12.1. Description

The **Smk2emis** program converts SMOKE I/O API gridded, mole-speciated, hourly emissions into the Fortran-binary format used for 2-D emissions by UAM, CAM_X, and REMSAD. Files of this format use the UAM keyword "EMISSIONS". The SMOKE user is responsible for preparing the I/O API inputs with

the appropriate chemical species, grid structures, time steps, and units for use in whatever model will be using the results of **Smk2emis**.

5.3.12.2. Processing Order

Smk2emis inputs the I/O API gridded, speciated, hourly emissions files output from **Smkmerge** (AGTS_L, MGTS_L, PGTS_L, or EGTS_L) or output from **Tmpbeis3** (BGTS_L).

5.3.12.3. Files and Environment Variables

5.3.12.3.1. Input Files

File Name	Format	Description
[A|B|M|P|E]GTS_L	I/O API NetCDF	SMOKE 2-D gridded, mole-speciated, hourly emissions file.
VNAMMAP	ASCII	Optional file for renaming variables from those in the I/O API file header to those needed by the AQM. See the SMK2EMIS_VMAP_YN environment variable below.

5.3.12.3.2. Input Environment Variables

- FLABEL: [default: EMISSIONS (currently the only working option)]

  Defines what type of UAM/REMSAD/CAM$_X$ file to output. This setting cannot be adjusted by the user, because the current version of **Smk2emis** can generate only the "EMISSIONS" file.

- SMK2EMIS_VMAP_YN: [default: N]

  Controls whether the name remapping file will be read in, to cause **Smk2emis** to output different variable names than are in the input file. This is needed for using UAM-AERO, which requires symbols in the variable names (such as ".", "+", and "-") that are not permitted in I/O API file variable names.

  - Y: Use the VNAMMAP file

  - N: Do not use the VNAMMAP file

- SMK_SOURCE: [default: none]

  Defines the type of sources to be processed. Valid values are:

  - A: Area sources

  - B: Biogenic sources

  - M: On-road mobile sources

  - P: Point sources

  - E: All emissions sources (determined by how **Smkmerge** was run)

  In **Smk2emis**, this variable controls the name of the input NetCDF logical file and the name of the output binary logical file. Unlike in most other SMOKE programs, for this program SMK_SOURCE can be set to "E" to indicate that the "every" source category merged file (EGTS_L) should be converted from I/O API to Fortran-binary format.

- UTM_ZONE: [default: -9]

Defines UTM zone for the header of the UAM EMISSIONS file.

- -9: Latitude-longitude projection

- `UAM_LAYERS`: [default: NLAYS3D]

  Defines number of layers in output file

  - NLAYS3D: Value from SMOKE netCDF input file

- `UAM4_LAYBELOW`: [default: 0]

  Defines number of layers below diffusion break

  - 0: UAM-V or CAM_X style

  - >0: UAM-IV style

- `UAM4_LAYABOVE`: [default: 0]

  Defines number of layers above diffusion break

  - 0: UAM-V or CAM_X style

    >0: UAM-IV style

- `UAM4_HTSFC`: [default: 0.0]

  Defines height of surface layer

  - 0.0: UAM-V or CAM_X style

  - >0.0: UAM-IV style

- `UAM4_HTLOWR`: [default: 0.0]

  Defines minimum height of cells between surface layer and diffusion break (meters)

  - 0.0: UAM-V or CAM_X style

  - >0.0: UAM-IV

- `UAM4_HTUPPR`: [default: 0.0]

  Defines minimum height of cells between diffusion break and top of region(meters)

  - 0.0: UAM-V or CAM_X style

  - >0.0: UAM-IV

- `UAM_NOTE`: [default: UAM gridded emissions from SMK2EMIS]

  Defines the note to be written to the header of the UAM, REMSAD, or CAM_X output file

5.3.12.3.3. Other Inputs

If you are running **Smk2emis** interactively, the following inputs will be requested. The default values for these are set by the time variables found in the input I/O API file header. If you are not running **Smk2emis**

interactively, the values of these settings will be taken from the input file being converted to Fortran-binary format, so no action from you is required.

- Start date: Starting date of simulation in Julian (DDDYYYY) or Gregorian (MMDDYYYY) formats

- Start Time: Starting time of simulation in HHMMSS.

- Duration: Simulation duration in hours.

5.3.12.3.4. Output Files

File Name	Format	Description
UAM_[A|B|M|P|E]GTS	Fortran binary	UAM-IV-, UAM-V- or CAMx-ready gridded, hourly emissions.
LOGFILE	ASCII	Log generated from executing **Smk2emis**.

5.3.13. Surgtool

5.3.13.1. Description

The **Surgtool** program inputs SMOKE-formatted gridding surrogates for a "fine" input grid and a grid definition for a "coarse" output grid. It produces an approximate "coarse" grid surrogates file, the accuracy of which depends on how fine the resolution of the input grid is relative to that of the output grid.

The program requires that the input and output grids use a lat-lon, Lambert Conformal, or UTM map projection. **Surgtool** can also perform Lambert-to-Lambert and UTM zone-to-zone transformations. In addition, the program requires that the input grid resolution be much finer than the output grid resolution (at least three times finer is recommended, though no specific limitation is given by the code). **Surgtool** supports SMOKE-formatted surrogate coefficient files only.

5.3.13.2. Processing Order

Since **Surgtool** is an optional program that creates new Area and/or Mobile BGPRO spatial surrogate files from existing Area, Mobile and/or Biogenic BGPRO spatial surrogate files, it must be run before any of the following programs that use these files: **Grdmat**, and **Smkmerge**.

5.3.13.3. Files and Environment Variables

5.3.13.3.1. Input Files

File Name	Format	Description
GRIDDESC	ASCII	Grid descriptions file that defines the output grid for the output surrogates file created by the program. It must include the name of the grid set by the IOAPI_GRIDNAME_1 environment variable setting.
INFILE	ASCII	Area and Mobile spatial surrogate file format (see Section 8.4.2, "Area and Mobile Spatial Surrogates Files" [387]) and Biogenic spatial surrogate file format (see Section 8.4.5, "BGPRO: Biogenic spatial surrogate file" [389]). File can be for Lambert Conformal, UTM, or lat-lon projection.

5.3.13.3.2. Input Environment Variables

- `IOAPI_GRIDNAME_1`: [default: blank]

    This variable sets the name of the output grid definition to be read from the `GRIDDESC` input file. The name can be a maximum of 16 characters (the limitation of I/O API grid names).

5.3.13.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Surgtool**
OUTFILE	ASCII	Estimated surrogates file for the grid defined in `GRIDDESC` and selected by the `IOAPI_GRIDNAME_1` setting. This surrogates file is not as accurate as creating a spatial surrogates file from the MIMS spatial allocator. However, the finer the **Surgtool** input grid is relative to the output grid, the more accurate the estimated surrogates are.

5.3.14. Uam2ncf

5.3.14.1. Description

The **Uam2ncf** program takes UAM (or REMSAD or CAM$_X$) EMISSIONS gridded files and creates an I/O API gridded file from them, which can then be merged with other gridded SMOKE I/O API files. This program is useful for converting area, nonroad mobile, on-road mobile, and biogenic UAM-ready files to the I/O API format needed for CMAQ or MAQSIP. **Uam2ncf** can also be used with low-level point sources, but not with the elevated-point-source inputs for UAM, REMSAD, and CAM$_X$.

If you wish to create CMAQ or MAQSIP inputs from model-ready emissions for UAM, REMSAD, or CAM$_X$, you must take the following steps:

1. Use **Uam2ncf** to convert individual source categories for area, biogenic, and mobile sources to I/O API format.

2. Run SMOKE for point-source inventory to create 3-D I/O API file.

3. Run **Mrggrid** to combine converted inventory for nonpoint sources with SMOKE-created inventory for point sources.

5.3.14.2. Processing Order

Because this optional program operates on model-ready files, it is run before the CMAQ or MAQSIP air quality model.

5.3.14.3. Files and Environment Variables

5.3.14.3.1. Input Files

File Name	Format	Description
GRIDDESC	ASCII	Grid descriptions file that defines the input grid in the `UAMEMIS` file. This is needed because the `UAMEMIS` file does not include enough information to describe the grid for the I/O API. The

File Name	Format	Description
		file must include the name of the grid set by the IOAPI_GRIDNAME_1 environment variable setting.
UAMEMIS	Fortran binary	UAM-ready, gridded, hourly, speciated emissions file.

5.3.14.3.2. Input Environment Variables

- IOAPI_GRIDNAME_1: [default: blank]

  This variable sets the name of the input (and output) grid definition to be read from the GRIDDESC input file. The name can be a maximum of 16 characters long (the limitation of I/O API grid names).

5.3.14.3.3. Output Files

File Name	Format	Description
E2DNCF:	I/O API NetCDF	Gridded, hourly, speciated emissions in the I/O API format needed for CMAQ or MAQSIP.
LOGFILE	ASCII	Log generated from executing **Uam2ncf**

Chapter 6. SMOKE Core Programs

Table of Contents

6.1. Overview

Figure 6.1, "SMOKE core programs" [212] shows the SMOKE core programs and how they relate to each other. We define the core programs as those needed for the actual emissions processing; we exclude QA and utility programs such as **Smkreport** (these programs are discussed in Chapter 7, *SMOKE Quality Assurance* [305] and Chapter 5, *SMOKE Utility Tools* [177]). The lines between the boxes represent one or more files passing between the programs, and show the dependencies among the programs. Note that we have omitted some redundant file dependencies to make the diagram easier to understand. Unless otherwise noted, the SMOKE core programs are applied for the anthropogenic source categories: area-, mobile-, and point-source processing (biogenic processing follows a separate path from the other source categories). When a program that applies to multiple source categories is used, it can be run for only one source category at a time, with the exception of **Smkmerge** and **Mrggrid**.

Figure 6.1. SMOKE core programs
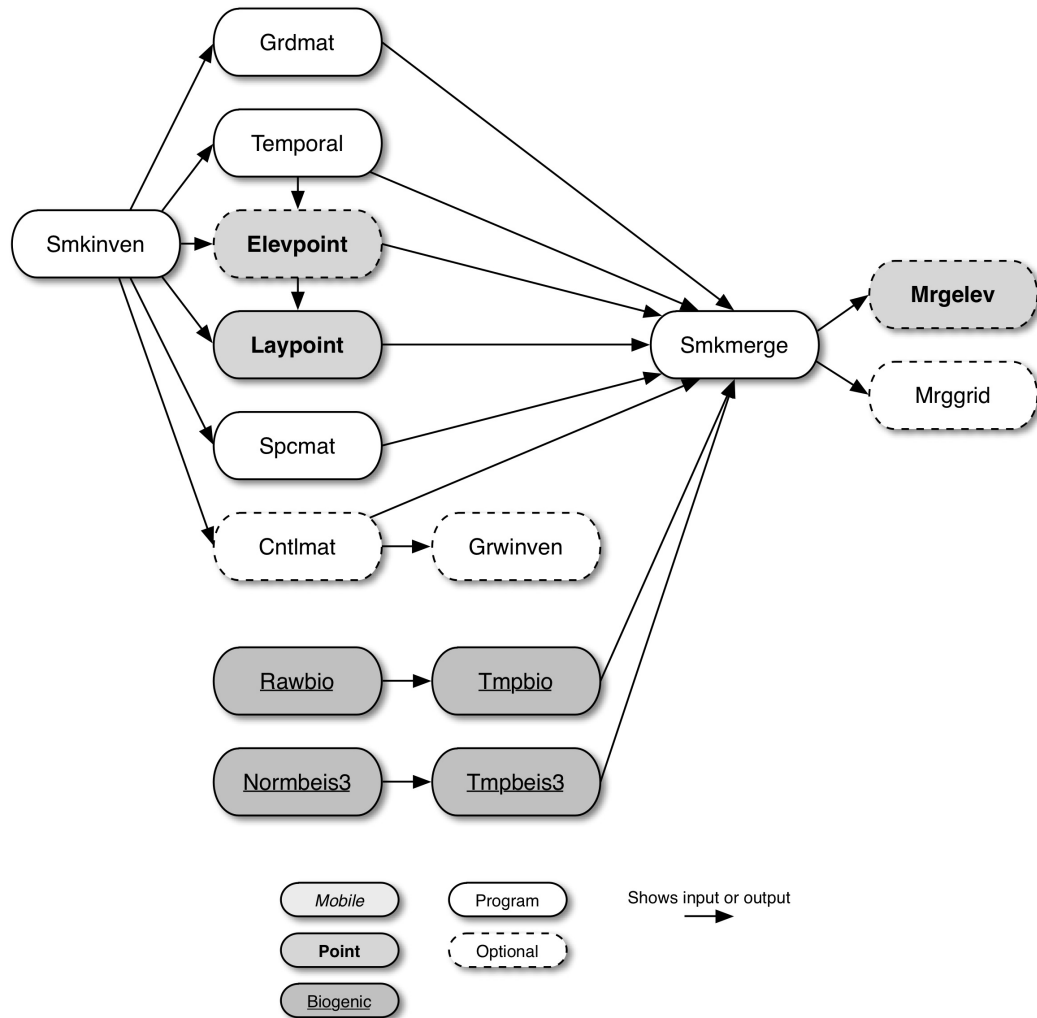
The **Smkinven** program, at left in Figure 6.1, "SMOKE core programs" [212], is responsible for importing the stationary area/non-point, nonroad, on-road mobile, and point source inventory emissions data. For mobile sources, **Smkinven** can also import activity data in the form of VMT and vehicle speed for use in generating emission factors. For point sources, **Smkinven** can also import day- and hour-specific data. The output from **Smkinven** is used as input to nearly every other core SMOKE program.

Also at left, the **Normbeis3** program imports the gridded land use data and biogenic emissions factors ,and creates gridded, normalized (time-independent) biogenic emissions.

Grdmat creates the gridding matrix for the anthropogenic source categories.

The **Temporal** program is used to create an hourly emissions file for the anthropogenic source categories. It can read in day-specific and hour-specific data, and merge this with estimated daily and hourly data created from the annual emissions data using temporal profiles.

Elevpoint preprocesses the selected PinG and major point sources. **Elevpoint** is not used when no PinG or major point sources need to be defined. In this case, SMOKE can create elevated emissions for all point sources, so there is no need to specifically indicate the major point sources.

Laypoint computes the plume rise for all point sources based on the meteorology data.

Spcmat creates the speciation matrices (both mass and molar) for the anthropogenic source categories. It uses the user-selected chemical mechanism.

Cntlmat creates the growth matrices, multiplicative control matrices, and reactivity control matrices for the anthropogenic source categories. This program is not used for base-year emissions without controls.

Grwinven is used to grow the emissions to past or future years using the growth matrix created by **Cntlmat** and the imported inventory data from **Smkinven**. It writes both SMOKE inventory format (I/O API NetCDF) and ORL format. The output from **Grwinven** is used in place of the original output from **Smkinven** when processing past or future years.

Tmpbeis3 applies meteorology adjustments to the normalized emissions created by **Normbeis3**. **Tmpbeis3** also applies the speciation profiles needed for the user-selected chemical mechanism to create gridded, hourly, model-species biogenic emissions data for use in **Smkmerge** and **Mrggrid**. The **Normbeis3** and **Tmpbeis3** programs taken together are the equivalent of SMOKE-BEIS3.

Smkmerge is used to combine all emissions and matrices to create the gridded, hourly, model-species emissions needed for an AQM. It can merge for one source category at a time or all source categories at once, and it can read in the model-ready biogenic emissions to merge with the anthropogenic source categories. It can merge the matrices with the inventory data output from **Smkinven** or the hourly emissions from **Temporal**, and it can optionally merge the speciation, gridding, or control matrices, or any combination. It also writes state and county emissions totals.

Movesmrg is only used for mobile sources to compute hourly emissions using the hourly VMT from **Temporal**, the MOVES emission rates lookup tables (i.e., RPD, RPH, RPV and RPP), the gridded/averaged temperatures meteorology data and more. If VMT are not used, then the emissions from **Smkinven** are combined with the temporal factors, and the emission factors are not used at all.

In the remaining sections of this chapter, we provide detailed program descriptions, in alphabetic order.

Mrgelev is used to combine ASCII elevated files created by **Smkmerge**. These files contain point-source data needed by AQMs including CAMx, REMSAD, and UAM models.

Mrggrid is used to combine gridded emission data files, which can be speciated or non-speciated, and hourly or time-independent. It can combine a 3-D point source file with any number of 2-D files from other source categories. This program is optional and provides a convenient way to merge model-ready output files outside of **Smkmerge**.

Mrgpt is used to combine multiple point sources inline-mode hourly speciated emission data and ASCII elevated files created by **Smkmerge**. This program is optional and allows the users to process multiple point sources separately in inline mode through **Smkmerge** prior to merging them for CMAQ model.

6.2. Cntlmat

6.2.1. Description

Processing categories: area, mobile, point

The **Cntlmat** program uses control packets to create a growth matrix, and/or a multiplicative control matrix, and/or a reactivity control matrix. **Cntlmat** allows seven different control packet types: /MACT/, /CONTROL/, /EMS_CONTROL/, /CTG/, /ALLOWABLE/, /REACTIVITY/, and /PROJECTION/. The format

of each packet type is given in Section 8.6.1, "GCNTL: Files for controls and/or past- and future-year growth" [395].

- Growth matrix - This contains past- or future-year growth factors for all sources.

- Multiplicative control matrix - This matrix contains the combination of various types of controls typically used in emissions processing: CTG, MACT, RACT, control efficiency, rule effectiveness, rule penetration, allowable emissions, and caps. The logic used when applying this matrix is described below this bulleted list.

- Reactivity matrix - This matrix contains the information needed for applying reactivity controls.

6.2.1.1. Application Logic for the Multiplicative Control Matrix

The multiplicative matrix is built from entries in four packets: /MACT/, /CONTROL/ or /EMS_CONTROL/, /CTG/, and /ALLOWABLE/. The packets are applied in the following order:

- /MACT/

- /CONTROL/ or /EMS_CONTROL/ (cannot use both at same time)

- /CTG/

- /ALLOWABLE/

With the exception of the /CONTROL/ packet, all controls are always applied in addition to one another.

6.2.1.1.1. /MACT/ packet

The /MACT/ packet is designed to support the primary MACT controls needed for toxics processing. It is applied first, and uses the following logic. When a source is found that matches a /MACT/ packet entry (source matching is addressed in Section 6.2.1.2, "Source Matching Hierarchy" [216]), **Cntlmat** first calculates a inventory efficiency value (Eff_{inv}) for that source based on the control efficiency (C_{eff}), rule effectiveness (R_{eff}), and rule penetration (R_{pen}) provided in the inventory:

$Eff_{inv} = C_{eff}$ [inventory] x R_{eff} [inventory] x R_{pen} [inventory]

Next, a control factor is calculated for existing sources (Fac_{exist}) using the control efficiency of existing sources (Eff_{exist}) and fraction of new sources ($Frac_{new}$) provided in the /MACT/ packet entry. If $Eff_{exist} > Eff_{inv}$, then Fac_{exist} is calculated as:

$Fac_{exist} = (1 - Frac_{new})$ x $((1 - Eff_{exist}) / (1 - Eff_{inv}))$

Otherwise,

$Fac_{exist} = (1 - Frac_{new})$

Cntlmat then calculates a control factor for new sources (Fac_{new}) using the control efficiency of new sources (Eff_{new}) provided in the /MACT/ packet. If $Eff_{new} > Eff_{inv}$, then

$Fac_{new} = Frac_{new}$ x $((1 - Eff_{new}) / (1 - Eff_{inv}))$

Otherwise,

$Fac_{new} = Frac_{new}$

Finally, the new and existing source factors are summed to compute the final /MACT/ packet control factor (Fac_{MACT}).

$$Fac_{MACT} = Fac_{exist} + Fac_{new}$$

The /MACT/ packet is designed to be used only for sources with MACT codes, currently nonpoint and point toxic sources.

6.2.1.1.2. /CONTROL/ packet

After processing the /MACT/ packet, the /CONTROL/ packet is applied using the following logic. Each entry in the /CONTROL/ packet can be identified as "replacement" or "additive". If set to "replace", the /CONTROL/ entry will be used to replace the controls from the /MACT/ packet or the base-year inventory controls; otherwise, the control will be applied in addition to the controls from the /MACT/ packet or the base-year inventory controls. When the controls are replaced, the default behaviour of **Cntlmat** is to back out the existing controls and apply the replacement controls. Usually, users want to do this only when the replacement controls are more stringent (i.e., a greater reduction) than the /MACT/ packet or base-year controls - so this is the default behaviour of **Cntlmat**. This default behaviour can be changed by setting the value of COMPARE_REPLACE_CONTROL to "N" in your run script. In this case, the replacement controls will be applied regardless of the existing controls from the base invetory or the /MACT/ packet, which could result in increased emissions.

When an "additive" entry is matched to a source, the /CONTROL/ packet factor ($Fac_{CONTROL}$) the control packet information is applied in addition to the exitsing controls, which further reduce the remaining emissions.

For "additive" control packet entries, the control factors are calculated from the control efficiency, rule effectiveness, and rule penetration values provided in the /CONTROL/ packet entry as follows:

$$Fac_{CONTROL} = (1 - C_{eff} [packet] \; x \; R_{eff} [packet] \; x \; R_{pen} [packet])$$

A cumulative control factor Fac_{result} is then calculated by multiplying $Fac_{Existing} \; x \; Fac_{CONTROL}$.

In the case of "additive" controls only, The overall reduction factor is then used to calculate an overall new control efficiency to be put in any resulting inventory files as: $C_{eff} [result] = 1 - Fac_{result}$. The final Rule Penetration and Rule Effectiveness values are then set to 100%.

For "replacement" entries, **Cntlmat** first "backs out" the existing controls provided in the inventory from the control values provided in the /CONTROL/ packet entry as follows:

$$D = (1 - C_{eff} [inventory] \; x \; R_{eff} [inventory] \; x \; R_{pen} [inventory])$$

If D = 0.0, then the backout value, B, is equal to 0; otherwise, B = 1/D. Using this backout value, the /CONTROL/ factor is calculated as

$$Fac_{CONTROL} = B \; x \; (1 - C_{eff} [packet] \; x \; R_{eff} [packet] \; x \; R_{pen} [packet])$$

Unlike "additive" entries, the $Fac_{CONTROL}$ value for "replacement" entries replaces the factor calculated from the /MACT/ packet or base year inventory (if any). The /CONTROL/ packet can be applied to stationary area/nonpoint, nonroad, on-road mobile, and point sources. On-road mobile source inventories do not include control efficiency, rule effectiveness, or rule penetration fields. However, the /CONTROL/ packet can still be used to apply adjustment factors to mobile sources. In this case, B in the above formula will be 1 when computing factor $Fac_{CONTROL}$ that is applied to the emissions.

6.2.1.1.3. /CTG/ packet

The /CTG/ packet is processed next, as follows. The emission value (after application of /MACT/, /CONTROL/, and/or /EMS CONTROL/) is compared to the cutoff value specified as part of the /CTG/ packet record. If the emission value exceeds the cutoff value, then the control technology factor is applied to the emissions. If the resulting emissions value still exceeds the cutoff value, then either the MACT *or* the RACT factor is used. The MACT factor is used if it is defined (i.e., greater than zero); if it is not defined (less than or equal to zero), then the RACT factor is used. If neither the MACT nor RACT are defined, but the emissions are still greater than the cutoff, then the emissions are set to the cutoff value.

6.2.1.1.4. /ALLOWABLE/ packet

Finally, the /ALLOWABLE/ packet is processed. The entries in this packet override the /MACT/, /CONTROL/, /EMS CONTROL/, and /CTG/ packets. The way this packet is applied depends on whether or not the Cap and Replace values in the packet are defined (greater than zero).

- If the Cap value is defined, **Cntlmat** compares the inventory emissions value to the Cap value, and sets a control factor, using the following procedure: If both the Cap and Replace values are defined, this control factor is the Replace value divided by the original emissions value. If only the Cap value is defined, the factor is the Cap value divided by the original emissions value.

- If the Cap value is not defined, then the factor is the Replace value divided by the original emissions value.

- If neither the Cap nor Replace values are defined, the packet entry is invalid and is ignored.

6.2.1.2. Source Matching Hierarchy

To match each sources with the appropriate control packet entry, **Cntlmat** uses a hierarchy of source characteristics, trying to match each source to the most specific control packet entry. Note that not all combinations are applicable to all control packets (for example, there is no MACT code associated with the /PROJECTION/ packet). "Left *x* of SCC" refers to an 8- or 10-digit SCC with all digits except the first *x* set to 0. "Left 2 of SIC code" refers to a 4-digit SIC code with the third and fourth digits set to 0. The maximum field width in SMOKE and its input files for SCC is 20 characters as of SMOKE v4.0. The 8 or 10 digit SCC are still supported, but if the SCC is greater than 10 digits the SCC hierarchial approach will not be supported. The SIC may also be up to 20 characters in length, but if the SIC is larger than 4 characters then the hierarchial approach will not be used.

For point sources, the following hierarchy is used, in order of most specific to least specific. (NOTE: only for SCC less than or equal to 10 characters and SIC less than or equal to 4 characters)

1. Country/State/County code, plant ID, point ID, stack ID, segment ID, 8-digit SCC code, pollutant
2. Country/State/County code, plant ID, point ID, stack ID, segment ID, pollutant
3. Country/State/County code, plant ID, point ID, stack ID, pollutant
4. Country/State/County code, plant ID, point ID, pollutant
5. Country/State/County code, plant ID, 8-digit SCC code, pollutant
6. Country/State/County code, plant ID, MACT code, pollutant
7. Country/State/County code, plant ID, pollutant
8. Country/State/County code, plant ID, point ID, stack ID, segment ID, 8-digit SCC code
9. Country/State/County code, plant ID, point ID, stack ID, segment ID
10. Country/State/County code, plant ID, point ID, stack ID
11. Country/State/County code, plant ID, point ID
12. Country/State/County code, plant ID, 8-digit SCC code
13. Country/State/County code, plant ID, MACT code

14. Country/State/County code, plant ID
15. Country/State/County code, MACT code, 8-digit SCC code, pollutant
16. Country/State/County code, MACT code, pollutant
17. Country/State code, MACT code, 8-digit SCC code, pollutant
18. Country/State code, MACT code, pollutant
19. MACT, 8-digit SCC code, pollutant
20. MACT, pollutant
21. Country/State/County code, MACT code, 8-digit SCC code
22. Country/State/County code, MACT code
23. Country/State code, MACT code, 8-digit SCC code
24. Country/State code, MACT code
25. MACT code, 8-digit SCC code
26. MACT code
27. Country/State/County code, 4-digit SIC code, pollutant
28. Country/State/County code, left 2 of SIC code, pollutant
29. Country/State code, 4-digit SIC code, pollutant
30. Country/State code, left 2 of SIC code, pollutant
31. 4-digit SIC code, pollutant
32. Left 2 of SIC code, pollutant
33. Country/State/County code, 4-digit SIC code
34. Country/State/County code, left 2 of SIC code
35. Country/State code, 4-digit SIC code
36. Country/State code, left 2 of SIC code
37. 4-digit SIC code
38. Left 2 of SIC code
39. Country/State/County code, 8-digit SCC code, pollutant
40. Country/State/County code, left 6 of SCC code, pollutant
41. Country/State/County code, left 3 of SCC code, pollutant
42. Country/State/County code, left 1 of SCC code, pollutant
43. Country/State code, 8-digit SCC code, pollutant
44. Country/State code, left 6 of SCC code, pollutant
45. Country/State code, left 3 of SCC code, pollutant
46. Country/State code, left 1 of SCC code, pollutant
47. 8-digit SCC code, pollutant
48. Left 6 of SCC code, pollutant
49. Left 3 of SCC code, pollutant
50. Left 1 of SCC code, pollutant
51. Country/State/County code, 8-digit SCC code
52. Country/State/County code, left 6 of SCC code
53. Country/State/County code, left 3 of SCC code
54. Country/State/County code, left 1 of SCC code
55. Country/State code, 8-digit SCC code
56. Country/State code, left 6 of SCC code
57. Country/State code, left 3 of SCC code
58. Country/State code, left 1 of SCC code
59. 8-digit SCC code
60. Left 6 of SCC code
61. Left 3 of SCC code
62. Left 1 of SCC code
63. Country/State/County code, pollutant
64. Country/State/County code
65. Country/State code, pollutant
66. Country/State code
67. Pollutant

For nonpoint/stationary area sources, the following hierarchy is used. The maximum field width in SMOKE and its input files for SCC is 20 characters as of SMOKE v4.0. The 8 or 10 digit SCC are still supported, but if the SCC is greater than 10 digits the SCC hierarchial approach will not be supported. The SIC may also be up to 20 characters in length, but if the SIC is larger than 4 characters then the hierarchial approach will not be used. Nonpoint SCCs can have either 8 or 10 digits, which is why there are two separate SCC levels mentioned for each of the SCC-based assignments.(NOTE: hierarchial approach is only for SCC less than or equal to 10 characters and SIC less than or equal to 4 characters)

1. Country/State/County code, MACT code, 8- or 10-digit SCC code, pollutant
2. Country/State/County code, MACT code, pollutant
3. Country/State code, MACT code, 8- or 10-digit SCC code, pollutant
4. Country/State code, MACT code, pollutant
5. MACT code, 8- or 10-digit SCC code, pollutant
6. MACT code, pollutant
7. Country/State/County code, MACT code, 8- or 10-digit SCC code
8. Country/State/County code, MACT code
9. Country/State code, MACT code, 8- or 10-digit SCC code
10. Country/State code, MACT code
11. MACT code, 8- or 10-digit SCC code
12. MACT code
13. Country/State/County code, 4-digit SIC code, pollutant
14. Country/State/County code, left 2 of SIC code, pollutant
15. Country/State code, 4-digit SIC code, pollutant
16. Country/State code, left 2 of SIC code, pollutant
17. 4-digit SIC code, pollutant
18. Left 2 of SIC code, pollutant
19. Country/State/County code, 4-digit SIC code
20. Country/State/County code, left 2 of SIC code
21. Country/State code, 4-digit SIC code
22. Country/State code, left 2 of SIC code
23. 4-digit SIC code
24. Left 2 of SIC code
25. Country/State/County code, 8- or 10-digit SCC code, pollutant
26. Country/State/County code, left 6 or left 7 of SCC code, pollutant
27. Country/State/County code, left 3 or left 4 of SCC code, pollutant
28. Country/State/County code, left 1 or left 2 of SCC code, pollutant
29. Country/State code, 8- or 10-digit SCC code, pollutant
30. Country/State code, left 6 or left 7 of SCC code, pollutant
31. Country/State code, left 3 or left 4 of SCC code, pollutant
32. Country/State code, left 1 or left 2 of SCC code, pollutant
33. 8- or 10-digit SCC code, pollutant
34. Left 6 or left 7 of SCC code, pollutant
35. Left 3 or left 4 of SCC code, pollutant
36. Left 1 or left 2 of SCC code, pollutant
37. Country/State/County code, 8- or 10-digit SCC code
38. Country/State/County code, left 6 or left 7 of SCC code
39. Country/State/County code, left 3 or left 4 of SCC code
40. Country/State/County code, left 1 or left 2 of SCC code
41. Country/State code, 8- or 10-digit SCC code
42. Country/State code, left 6 or left 7 of SCC code
43. Country/State code, left 3 or left 4 of SCC code
44. Country/State code, left 1 or left 2 of SCC code
45. 8- or 10-digit SCC code
46. Left 6 or left 7 of SCC code

47. Left 3 or left 4 of SCC code
48. Left 1 or left 2 of SCC code
49. Country/State/County code, pollutant
50. Country/State/County code
51. Country/State code, pollutant
52. Country/State code
53. Pollutant

On-road mobile and nonroad sources are matched to the control packet entries using the following hierarchy if the SCC length is less than 10 characters. As of SMOKE v4.0 the SCC character length may be up to 20 characters in length:(NOTE: this hierarchial approach is only for SCC less than or equal to 10 characters and SIC less than or equal to 4 characters)

1. Country/State/County code, 10-digit SCC code, pollutant
2. Country/State/County code, left 7 of SCC code, pollutant
3. Country/State/County code, left 4 of SCC code, pollutant
4. Country/State/County code, left 2 of SCC code, pollutant
5. Country/State code, 10-digit SCC code, pollutant
6. Country/State code, left 7 of SCC code, pollutant
7. Country/State code, left 4 of SCC code, pollutant
8. Country/State code, left 2 of SCC code, pollutant
9. 10-digit SCC code, pollutant
10. Left 7 of SCC code, pollutant
11. Left 4 of SCC code, pollutant
12. Left 2 of SCC code, pollutant
13. Country/State/County code, 10-digit SCC code
14. Country/State/County code, left 7 of SCC code
15. Country/State/County code, left 4 of SCC code
16. Country/State/County code, left 2 of SCC code
17. Country/State code, 10-digit SCC code
18. Country/State code, left 7 of SCC code
19. Country/State code, left 4 of SCC code
20. Country/State code, left 2 of SCC code
21. 10-digit SCC code
22. Left 7 of SCC code
23. Left 4 of SCC code
24. Left 2 of SCC code
25. Country/State/County code, pollutant
26. Country/State/County code
27. Country/State code, pollutant
28. Country/State code
29. Pollutant

6.2.2. Processing Order

Smkinven must be executed before **Cntlmat**.

Cntlmat must be executed before **Smkmerge**. **Cntlmat** also needs to be run prior to the **Grwinven** program, which requires at least one growth or control matrix to be run.

Cntlmat is an optional program that needs to be run only when controlled emissions are needed for the AQM.

6.2.3. Files and Environment Variables

Figure 6.2. Cntlmat input and output files

Figure 6.2, "**Cntlmat** input and output files" [220] shows the input and output files for the **Cntlmat** program. The input files are the inventory file output from **Smkinven** (either `AREA`, `MOBL`, or `PNTS`), the control packets file (`GCNTL`), and the speciation profiles file (`GSPRO`) if there are reactivity controls (using the /REACTIVITY/ packet).

If the `GCNTL` file contains /MACT/, /CONTROL/ or /EMS_CONTROL/, /CTG/, and/or /ALLOWABLE/ packets, **Cntlmat** outputs a multiplicative control matrix file (`ACMAT`, `MCMAT`, or `PCMAT`), a report of multiplicative controls (`ACREP`, `MCREP`, or `PCREP`), and a report of all multiplicative controls applied to the inventory (`ACSUMREP`, `MCSUMREP`, or `PCSUMREP`). **Cntlmat** also outputs a warning file (`ACTLWARN`, `MCTLWARN`, or `PCTLWARN`) regardless of which packets are used.

If the `GCNTL` file contains a /PROJECTION/ packet, **Cntlmat** outputs a growth matrix file (`APMAT`, `MPMAT`, or `PPMAT`) and a growth report (`APROJREP`, `MPROJREP`, or `PPROJREP`).

If the `GCNTL` file contains a /REACTIVITY/ packet, **Cntlmat** outputs mole and mass-based reactivity matrix files (either `ARMAT_L` and `ARMAT_S`, `MRMAT_L` and `MRMAT_S`, or `PRMAT_L` and `PRMAT_S`), a reactivity matrix supplement file (`ARSUP`, `MRSUP`, or `PRSUP`), and a report of reactivity controls (`AREACREP`, `MREACREP`, or `PREACREP`). Finally, **Cntlmat** writes the `LOGFILE` detailing the program's execution.

6.2.3.1. Input Files

File Name*	Format	Description
`[AREA|MOBL|PNTS]`	SMOKE map	Intermediate inventory file produced by **Smkinven**
`GCNTL`	ASCII	Control packets file
`GSPRO` (optional)	ASCII	Speciation profiles file; only needed for reactivity controls

* As in Figure 6.2, "**Cntlmat** input and output files" [220], the notation `[AREA|MOBL|PNTS]` is used to indicate that only one of the three files is used for a given run.

The physical file name of the input file `GCNTL` file is typically overridden by the environment that is running **Cntlmat**. For example, when running SMOKE from scripts using the Assigns file, the projection factor inputs are available as the `[A|M|P]PROJ` files. So, for instance, the script must set `GCNTL` as `APROJ` (or area-source processing) after the Assigns file has been invoked by the script, but before the **Cntlmat** program is actually run.

6.2.3.2. Input Environment Variables

- `COMPARE_REPLACE_CONTROL:` [default: Y]

  Controls the way a replacement control is applied.

  - Y: When set to Y, Cntlmat compares the overall percent reductions (CEFF * RPEN * REFF) from a replacement control in the CONTROL packet against existing percent reductions (either from the base inventory or from a just-applied /MACT/ packet). If the packet's reduction is more control (greater reduction) than the existing controls, then it is applied. Otherwise, it is not applied.

  - N: When set to N, Cntlmat applies replacement controls even if the control is less control than an existing control, which may result in an increase in emissions

- `FULLSCC_ONLY:` [default: N]

  Disables any sub-SCC cross-reference assignments that would otherwise be included in the cross-reference hierarchy. For example, the left-7, left-4, and left-2 levels of nonpoint and mobile SCCs would never be used. This is useful when zeros are a natural part of the SCC that the user does not want to assume are present for cross-reference hierarchy purposes. This option affects all Cntlmat input packets that use SCC assignments as part of their hierarchy.

- `PROJECTION_YR_SPEC:` [default: Y]

  Sets whether or not entries in the /PROJECTION/ packet are in year-specific format.

  - Y: Year-specific format in use

  - N: Year-specific format not in use

- `REACTIVITY_POL`: [default: VOC]

  Sets the pollutant name to use in computing the reactivity matrix.

- `REPORT_DEFAULTS`: [default: Y]

  Defines whether or not the program reports when a source uses the default cross-reference.

  - Y: List sources in the log file when they use default cross-reference

  - N: Do not list sources that use the default cross-reference

- `SMK_AVEDAY_YN`: [default: N]

  Indicates whether or not to read average day emissions instead of annual emissions from the inventory file.

  - Y: Read average day emissions

  - N: Do not read average day emissions, use annual emissions instead

- `SMK_SOURCE`: [default: blank]

  Defines the type of sources to be processed. Valid values are

  - A: Area processing

  - B: Biogenic processing

  - M: Mobile processing

  - P: Point processing

- `SMK_TMPDIR`: [default: "."]

  Sets the path for writing the temporary files needed when processing the /ALLOWABLE/, /CONTROL/, /CTG/, /EMS_CONTROL/, /MACT/, and/or /PROJECTION/ packets (from the `GCNTL` file).

- `SPEC_OUTPUT`: [default: ALL]

  Defines the types of speciation outputs to create.

  - MASS: Output speciation matrix based on mass

  - MOLE: Output speciation matrix based on moles

  - ALL: Output both mass and mole speciation matrices

- `XREF_SICOVERSCC`: [default: Y]

  Determines how to deal with packet entries that have both an SCC and SIC

  - Y: Match sources by SIC before SCC

  - N: Match sources by SCC before SIC

6.2.3.3. Output Files

File Name*	Format	Description
`[A|M|P]CMAT`	I/O API NetCDF	Multiplicative control matrix
`[A|M|P]CREP`	ASCII	Report of multiplicative controls applied to the inventory
`[A|M|P]CREP2`	ASCII	Detailed before-and-after report of application of the /CONTROL/ packet only, which includes any comments appended to the end of the /CONTROL/ packet entries. These comments can describe the controls being applied with each packet row. This report then includes the base and future overall percent reductions on a source-by-source basis, and includes the CONTROL packet line-specific comments.
`[A|M|P]CSUM-REP`	ASCII	Report of all controls applied to the inventory, with all controls listed under each source
`[A|M|P]CTL-WARN`	ASCII	List of warnings
`[A|M|P]PMAT`	I/O API NetCDF	Growth matrix
`[A|M|P]PROJREP`	ASCII	Report of growth factors applied to the inventory
`[A|M|P]REACREP`	ASCII	Report of reactivity controls applied to the inventory
`[A|M|P]RMAT_L`	I/O API NetCDF	Mole-based reactivity matrix
`[A|M|P]RMAT_S`	I/O API NetCDF	Mass-based reactivity matrix
`[A|M|P]RSUP`	ASCII	Reactivity matrix supplemental file
`LOGFILE`	ASCII	Log generated from executing **Cntlmat**

* As in Figure 6.2, "**Cntlmat** input and output files" [220], the notation `[A|M|P]` is used to indicate that only one of the three files is output for a given run.

6.3. Elevpoint

6.3.1. Description

Processing categories: point

The **Elevpoint** program is used to select elevated point sources and to prepare certain input files for special elevated source or PinG processing. **Elevpoint** selects the major point source (MPS) and major elevated point source emissions (MEPSE) records using a variety of criteria, such as stack parameters, emissions, and analytical plume rise calculations. (Note that MEPSE is another name for a PinG source.)

Elevpoint distinguishes PinG sources from non-PinG elevated sources based on emission thresholds, highest day-specific emissions rank compared to other sources, analytical plume rise, and/or stack parameters such as height. **Elevpoint** flags sources as either elevated or PinG based on instructions given in an input configuration file, the `PELVCONFIG` file. **Elevpoint** also permits grouping of stacks at the same facility if the stack parameters are similar, within tolerances you specify with the `PELVCONFIG` file; the result is call a "stack group". For more information on selecting and grouping elevated and PinG sources, refer to Section 8.9.3.1, "`PELVCONFIG`: Elevated source selection file" [439].

Elevpoint can also be used to calculate plume rise using the "cutoff" method. This is the method traditionally used to prepare input files to the air quality models UAM-IV, UAM-V, REMSAD, and CAM$_X$, which require

a separate model-ready input file containing emissions for elevated point sources. Below, we refer to this path as the "UAM style" of emissions processing.

For the cutoff method, the **Elevpoint** program uses the stack parameters of the point sources and estimates the plume rise for each source using a Briggs analytical solution (but not actual gridded meteorology data); the details of this process are given in Section 6.3.1.1, "Plume rise calculation" [224]. **Elevpoint** then records all of the sources that have an estimated plume rise greater than a specified cutoff as defined in the `PEL-VCONFIG` file using the "RISE" instruction. Using the `PELVCONFIG` file, this cutoff method can be combined with selection of elevated sources based on stack parameters and/or emissions values.

When emissions values are part of these selection criteria, **Elevpoint** computes the maximum daily emissions using one or more `PTMP` files. **Elevpoint** computes the maximum daily value of any pollutant used as a selection criterion across all of the days that you input. You have the option of providing either a list of paths and file names of one or more `PTMP` files, or a single `PTMP` file. The day or days that **Elevpoint** uses to compute each source's maximum depends on the time zone in which the source resides, based on the county and the `COSTCY` file setting for the time zone (the time zone is stored in the `PNTS` file output from **Smkinven**). When using emissions values as a selection criterion, **Elevpoint** only considers sources that are within the grid domain.

Elevpoint also creates a report file that contains all SMOKE sources that have been selected as elevated, their source characteristics (including stack parameters), emissions values used as selection criteria, stack-group numbers, status as either elevated or PinG, and which selection criteria were used to assign the status to the source. The report does not include any low-level sources. This report is in semicolon-delimited format for easier reading into a spreadsheet for sorting, review, and making charts.

The **Elevpoint** program is not needed if the environment variables `SMK_PING_METHOD` and `SMK_EL-EV_METHOD` are both set to 0.

6.3.1.1. Plume rise calculation

The Briggs solution is documented in the following:

1. Briggs, Gary A., 1971: Some Recent Analyses of Plume Rise Observation, pp. 1029-1032 in *Proceedings of the Second International Clean Air Congress*, edited by H. M. Englun and W. T. Beery. Academic Press, New York.

2. Briggs, Gary A., 1972: Discussion on Chimney Plumes in Neutral and Stable Surroundings, *Atmos. Environ.* 6, 507-510.

The following formula is used in **Elevpoint**:

$F = 0.25 \times G \times V_S \times D_S^2 \times (T_S - T)/T_S$

For F less than 55,

Plume rise $= H_S + 21.31311057 \times F^{0.75}/U$

Otherwise,

Plume rise $= H_S + 38.87776061 \times F^{0.6}/U$

where

- F = Bouyancy flux (m^4/s^3)

- G = Mean gravitational acceleration (9.80665 m/s^2)

- V_S = Stack gas exit velocity (m/s)

- D_S = Inside stack diameter (m)

- T_S = Stack gas temperature (K)

- T = Default ambient air temperature (293 K)

- U = Default wind speed (2 m/s)

- H_S = Physical stack height (m)

This represents an improved algorithm over the original so that the discontinuity at F = 55 no longer exists. Now, the second term in the plume rise formula approaches 595 as F approaches 55 from either side.

6.3.1.2. Summary

In summary, **Elevpoint** permits the following combinations of source selection and processing:

- Identify MEPSE/PinG sources for outputting to special PinG files only (all other point sources will be treated as potentially elevated by **Laypoint**).

- Identify elevated sources (MPSs) with the PELVCONFIG file, which might include using the cutoff method alone or in combination with other criteria.

- Identify elevated sources (MPSs) and MEPSE/PinG sources with the PELVCONFIG file with cutoff method (**Laypoint** not used) and identify MEPSE/PinG sources with the PSPLIT file for outputting to special PinG files.

6.3.2. Processing Order

Smkinven must be run for point sources before **Elevpoint**. If the PELVCONFIG file uses emissions as a source-grouping or source-selection criterion, then **Temporal** must be run before **Elevpoint**.

Elevpoint must be run before **Laypoint**.

Elevpoint must be run before **Smkmerge** if elevated sources are to be used.

Elevpoint is an optional program that is not needed if all of the following criteria are met:

1. A UAM-style model (e.g., UAM, REMSAD, CAM$_X$) is not being used.

2. No MEPSE/PinG source processing is required.

3. You are willing to run **Laypoint** for all point sources and treat all as "potentially elevated". In this case, sources that rise above layer 1 will be "elevated" for reporting purposes in **Smkreport**, but these sources may sometimes be within layer 1 if the plume rise algorithm in **Laypoint** computes it that way.

6.3.3. Files and Environment Variables
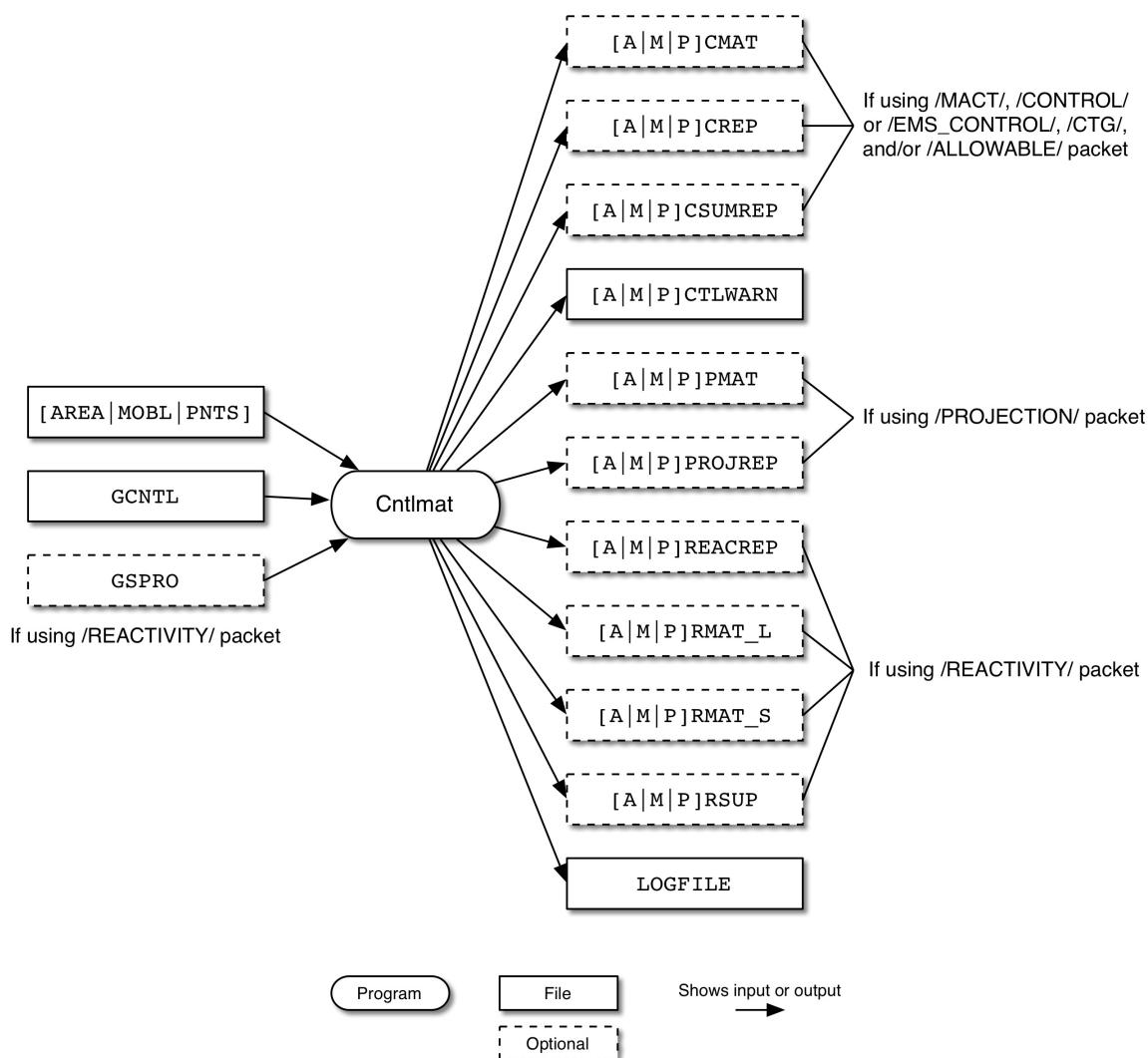
Figure 6.3. Elevpoint input and output files
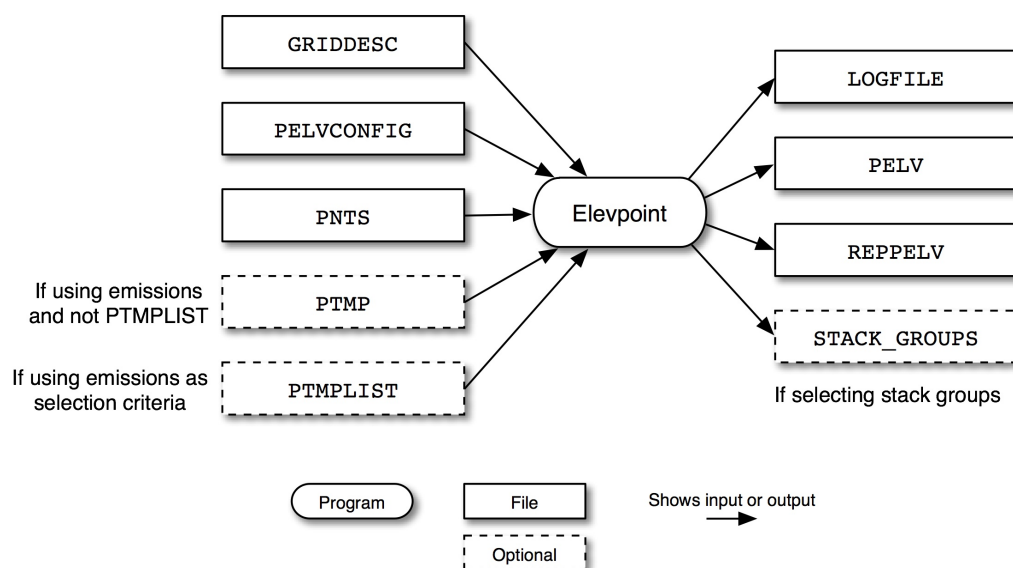
Figure 6.3, "**Elevpoint** input and output files" [226] shows the input and output files for the **Elevpoint** program. The input files are the grid description file (GRIDDESC), an elevated source configuration file (PELVCONFIG), the point-source inventory file output from **Smkinven** (PNTS), and, if the PELVCONFIG file selects sources by emissions value or rank, either a list of hourly emissions files output by **Temporal** (PTMPLIST) or an individual hourly emissions file (PTMP).

Elevpoint outputs a log file (LOGFILE), an elevated point source identification file (PELV), an elevated and PinG source report (REPPELV), and a stack parameters file (STACK_GROUPS) if the PELVCONFIG file is configured to select stack groups.

6.3.3.1. Input Files

File Name	Format	Description
GRIDDESC	ASCII	Description of the modeling grid
PELVCONFIG	ASCII	Elevated source configuration file
PNTS	SMOKE map	Intermediate inventory file produced by **Smkinven**
PTMP (optional)	ASCII	Hourly emissions file produced by **Temporal**; needed only when PELVCONFIG specifies emissions as one of the selection or grouping criteria and not using the PTMPLIST file
PTMPLIST (optional)	ASCII	List of hourly emissions files produced by **Temporal**; needed only when PELVCONFIG specifies emissions as one of the selection or grouping criteria

6.3.3.2. Input Environment Variables

- ELEV_WRITE_LATLON: [default: Y]

Defines whether write the latitude and longitude information of special elevated or PinG sources to the stack groups file Section 9.3.2, "STACK_GROUPS" [464] for air quality models CMAQ, MAQSIP, and UAM-style.

- Y: Write the latitude and longitude information of elevated or PinG sources to the the stack group file.

- N: Do not write the latitude and longitude information of elevated or PinG sources to the the stack group file.

The ELEV_WRITE_LATLON can not be N, when processing for PinG sources, since CMAQ does need the latitude and longitude to be included.

- G_RUNLEN: [default: 240000]

  Simulation length (HHMMSS)

- G_STDATE: [default: 0]

  Starting date of simulation (YYYYDDD)

- G_STTIME: [default: 0]

  Starting time of simulation (HHMMSS)

- G_TSTEP: [default: 10000]

  Time step (HHMMSS), only 10000 is allowed

- IOAPI_ISPH: [default: 19]

  Determines which USGS spheroid to use for grid-based calculations
  - 0: Clarke 1866
  - 1: Clarke 1880
  - 2: Bessel
  - 3: New International 1967
  - 4: International 1909
  - 5: WGS 72
  - 6: Everest
  - 7: WGS 66
  - 8: GRS 1980
  - 9: Airy
  - 10: Modified Everest
  - 11: Modified Airy
  - 12: WGS 84
  - 13: Southeast Asia
  - 14: Australian National
  - 15: Krassovsky
  - 16: Hough
  - 17: Mercury 1960
  - 18: Modified Mercury 1968
  - 19: Normal Sphere

  This environment variable can also be used to set custom spheroid representations. See the I/O API documentation [http://www.baronams.com/products/ioapi/SETSPHERE.html] for more information.

- `SMK_ELEV_METHOD`: [default: 0]

  Defines the method to use in SMOKE for determining and processing elevated point sources. The setting chosen depends on the model output format needed from SMOKE. All PinG sources determined based on `SMK_PING_METHOD` will automatically be considered elevated sources (as defined specifically for the model for which you are creating emissions).

  - 0: Treat all sources as potentially elevated, and let the **Laypoint** program determine which sources are elevated and which are not. This approach is appropriate only for SMOKE output for the CMAQ and MAQSIP models.

  - 1: Use the "Elevated" section of the `PELVCONFIG` configuration file to select specific sources. This approach can be used for any of the model output formats - CMAQ, MAQSIP, and UAM-style.

    For UAM-style emissions processing, this environment variable must be set to 1 and the `PELVCON-FIG` file will be used to select specific sources. This method of processing is needed for UAM, REMSAD, and CAM$_X$ because they require use of the cutoff method discussed earlier. In rare cases of a very small inventory, it might be acceptable to specify 0 for this setting in UAM-style emissions processing, provided that you are sure that treating all sources as elevated sources in UAM-style models is acceptable. We do not recommend such an approach for typical SMOKE applications for UAM-style models.

  - 2: Use the "Elevated" section of the `PELVCONFIG` configuration file to select specific sources. This approach can be used for in-line plume rise in CMAQ. Must set `SMK_PING_METHOD` = 0.

    For in-line plume rise processing in CMAQ, this environment variable must be set to 2 and the `PELVCONFIG` file will be used to select specific sources. To reduce the total number of plume rise calculations that would be done in the CMAQ in-line plume rise module, **Elevpoint** needs to group sources with nearly identical stack parameters. **Elevpoint** outputs stack group file that identifies elevated source locations and stack parameters for in-line plume rise calculation in CMAQ. See Section 4.4.18, "In-line Plume Rise Calculation in CMAQ model (as of version 4.7)" [169] for more detail.

    When using **Smkmerge** for in-line plume rise processing, `MRG_LAYERS_YN` must set to N to create hourly speciated 2-D surface layer emissions file. If `SMKMERGE_CUSTOM_OUTPUT` is set to Y, the output file for the in-line emissions is called `INLNST_L`.

    To process point fire emission (ptfire sector) without precomputed plume rise described in described in Section 4.4.2.5, "Day-specific and hour-specific point and fire inventories" [136].

- `SMK_PING_METHOD`: [default: 0]

  Defines the method to use in determining PinG point sources.

  - 0: No PinG sources

  - 1: Use PinG section of `PELVCONFIG` configuration file

  For all models that SMOKE supports except MAQSIP, use of PinG sources is optional. MAQSIP does not include a released PinG implementation. Some models may have limitations on the number of PinG sources supported, and you must make sure that these limitations are met; SMOKE does not automatically compare with these limitations, if any.

  For CMAQ, SMOKE will output a `STACK_GROUPS` file from the **Elevpoint** program and a `PINGTS_[S|L]` file from the **Smkmerge** program to support PinG. For UAM-style processing,

SMOKE will flag PinG sources in the ASCII elevated file created by **Smkmerge**, and should be used with the `SMK_ASCIIELEV_YN` option set to Y to create this ASCII elevated file.

- `SMK_SOURCE`: See description in Section 6.2.3.2, "Input Environment Variables" [221]

6.3.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Elevpoint**
PELV	ASCII	Elevated point source identification file
REPPELV	ASCII	Reports the sources selected as elevated and PinG, plus other information about the sources, including why they were selected
STACK_GROUPS	I/O API NetCDF	PinG and/or elevated sources stack parameters file

6.4. Grdmat

6.4.1. Description

Processing categories: area, mobile, point

The **Grdmat** program produces the matrix that contains the factors for spatially allocating the emission sources to the modeling domain.

For pre-gridded inventory data, grdmat is used to convert the data from the input data grid (e.g. lat-lon for EDGAR) to the output modeling grid when the user sets IMPORT_GRDNETCDF_YN to Y. Spatial surrogates (AGPRO or MGPRO) are not needed for processing gridded inventory data with SMOKE. The input modeling domain information is obtained from the GRIDMASK input file and the output grid is defined by the IOAPI_GRIDNAME_1 environment variable in the GRID_CRO_2D file.

For inventory data that is not already pre-gridded, point sources are assigned to grid cells using the geographic coordinates that are stored in each point-source record. For area and mobile sources, a cross-reference file matches the gridding surrogates to the source-level emissions. When the gridding cross-reference files are applied, the most-specific matches always override the less-specific ones.

For the current version of SMOKE, a gridding surrogates file must be created for the specific grid of interest. This can be accomplished in a variety of ways:

- ArcInfo (or other GIS) processing by an expert in creating gridding surrogates.

- Using the MIMS Spatial Allocation Tool.

- Extracting surrogate data from a larger gridding surrogates file. National 4-km surrogates are available from the EPA at http://www.epa.gov/ttn/chief/emch/spatial/newsurrogate.html. To use these data for your modeling, you must use the same grid projection as was used to create the gridding surrogates.

For area sources, the gridding cross-reference file provides the following possible combinations, ordered from most specific to least specific:

1. Country/state/county code and 10-digit SCC
2. Country/state/county code and 7-digit SCC
3. Country/state code and 10-digit SCC
4. Country/state code and 7-digit SCC

5. 10-digit SCC
6. 7-digit SCC
7. Country/state/county code
8. Country/state code
9. Ultimate default (country/state/county code = 0, SCC = 0)

For mobile sources, the available combinations and their order are:

1. Country/state/county code, and 10-digit SCC
2. Country/state code and 10-digit SCC
3. Country/state/county code
4. Country/state code
5. 10-digit SCC
6. Ultimate default (country/state/county code = 0, SCC = 0)

6.4.2. Processing Order

Smkinven must be run before **Grdmat**.

Grdmat must be run before gridded emissions may be produced using **Smkmerge**.

6.4.3. Files and Environment Variables

Figure 6.4. Grdmat input and output files
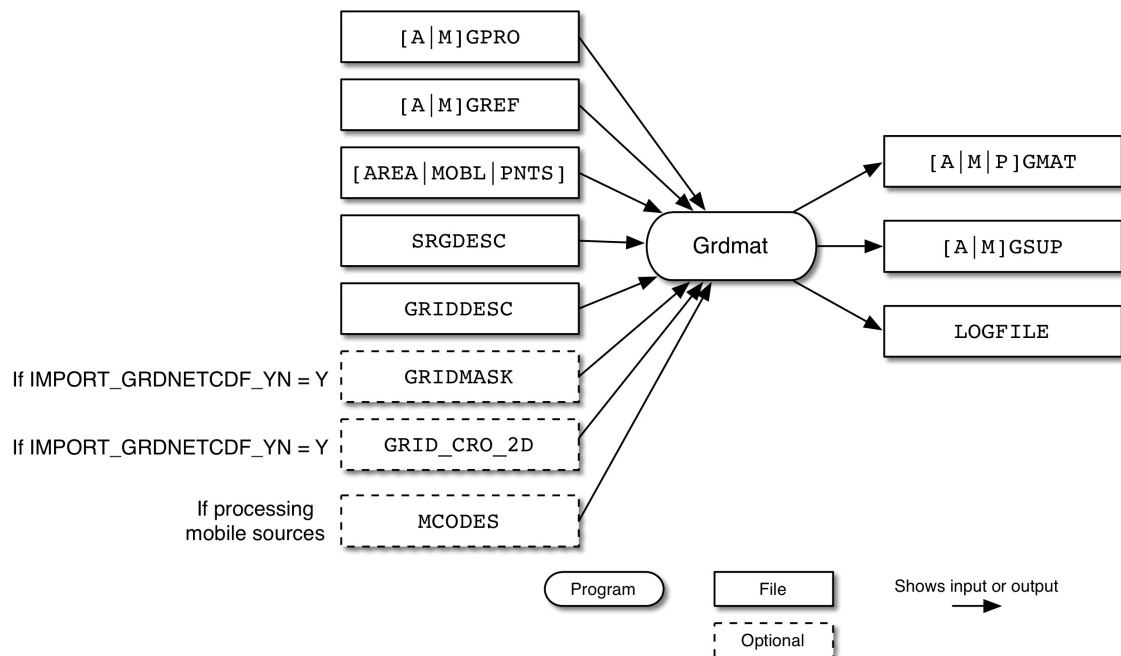
Figure 6.4, "**Grdmat** input and output files" [230] shows the input and output files for the **Grdmat** program. The input files are the spatial surrogates file (`AGPRO` or `MGPRO`), the spatial cross-reference file (`AGREF` or `MGREF`), the inventory file output from **Smkinven** (`AREA`, `MOBL`, or `PNTS`), the surrogate description file (`SRGDESC`), and the grid description file (`GRIDDESC`).

Grdmat outputs a gridding matrix file (AGMAT, MGMAT, or PGMAT) used by other SMOKE programs, and a supplemental gridding file (AGSUP or MGSUP) that includes all of the spatial surrogate codes assigned to each source. Like all SMOKE programs, **Grdmat** writes a program log (LOGFILE).

6.4.3.1. Input Files

File Name	Format	Description
SRGDESC	ASCII	Surrogate list, description and surrogate file name used in the modeling grid. Note: we recommand to use multiple files for individual surrogate code which are available from the EPA's New Surrogates web site [http://www.epa.gov/ttn/chief/emch/spatial/newsurrogate.html].
SRGPRO_PATH/surrogate-files	ASCII	Gridding surrogate coefficients
[A|M]GREF	ASCII	Gridding surrogate cross-reference file
[AREA|MOBL|PNTS]	SMOKE map	Intermediate inventory file produced by **Smkinven**
GRIDDESC	ASCII	Description of the modeling grid
GRIDMASK	I/O API NetCDF	Geographic level code and Timezone for each grid cell
GRID_CRO_2D	I/O API NetCDF	Layer definitions for the CMAQ model domain

6.4.3.2. Input Environment Variables

- FULLSCC_ONLY: See description in Section 6.2.3.2, "Input Environment Variables" [221]. In this context, this option prevents left-7 SCC assignments using the gridding cross-reference file.

- IOAPI_GRIDNAME_1: output modeling grid that global gridded inventory data is re-gridded to.

- IOAPI_ISPH: See description in Section 6.3.3.2, "Input Environment Variables" [226]

- IMPORT_GRDNETCDF_YN: Y to read the native netCDF annual inventory files

- REPORT_DEFAULTS: See description in Section 6.2.3.2, "Input Environment Variables" [221]

- SRGPRO_PATH: Defines the location of the Area and Mobile spatial surrogate files. This should be a sub-directory in the $GE_DAT directory.

- SMK_DEFAULT_SRGID: [default: 100 (Models-3 population surrogate code)] Mandatory, user MUST choose a fallback code.

- SMK_USE_FALLBACK: [Y/N]

  Defines the surrogate code number to use as a fallback surrogate when the surrogate number set by the cross-reference file would zero out the emissions for that source. This fallback surrogate is also used to define the county/state/country (FIPS) codes for use in **Grdmat**; therefore, a fallback surrogate code is mandatory even if the SMK_USE_FALLBACK is set to [N]. This code should exist in the SRGDESC file and be representative on all county/state/countries. See description in Section 8.4.1, "SRGDESC: Surrogate Designation, Description and Surrogate profile filename." [387].

- SMK_SOURCE: See description in Section 6.2.3.2, "Input Environment Variables" [221]

6.4.3.3. Output Files

File Name	Format	Description
[A|M|P]GMAT	I/O API NetCDF	Gridding matrix containing the factors needed to spatially allocate the input emissions to the grid cells in the modeling domain.
[A|M]GSUP	ASCII	Gridding supplemental file containing spatial surrogate codes for each source
LOGFILE	ASCII	Log generated from executing **Grdmat**

6.5. Grwinven

6.5.1. Description

Processing catagories: area, mobile, point

The **Grwinven** program grows and controls inventory emissions and activity data into the past/future by merging a SMOKE inventory file (AREA, MOBL, or PNTS) with its respective growth and control matrices. This optional program need not be used if these projections have been made outside of SMOKE. **Grwinven** outputs new SMOKE I/O API inventory files, but not the corresponding inventory ASCII files (because the program changes the data only in the I/O API inventory files, so there is no need to output the same ASCII files). The program can also output data in ORL format, or be used as a converter from SMOKE to ORL format. **Grwinven** does not merge the SMOKE inventory with reactivity matrices; this functionality is handled by **Smkmerge**.

The **Grwinven** program writes out temporary emissions files when creating an ORL-formatted inventory. You can set the directory to which these temporary files go by setting the SMK_TMPDIR environment variable. These files can be quite large; at least 100 MB should be available on the disk to which this environment variable is set.

6.5.2. Processing Order

Smkinven and **Cntlmat** must be run before **Grwinven**.

After running **Grwinven**, the output can be used similarly to the output from **Smkinven**. If **Spcmat** or **Grdmat** have already been run for a given inventory, these programs do not need to be rerun after running **Grwinven** because the number and order of the sources in the grown emissions files are the same as in the base case.

6.5.3. Files and Environment Variables

Figure 6.5. Grwinven input and output files
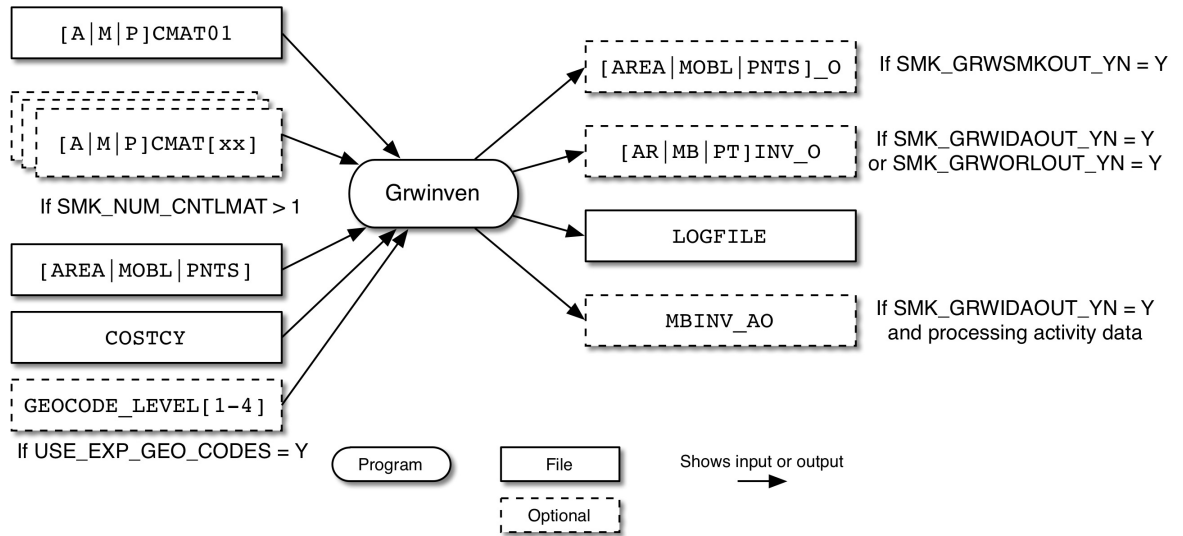
Figure 6.5, "**Grwinven** input and output files" [233] shows the input and output files for the **Grwinven** program. The input files include at least one growth or control matrix output from **Cntlmat** (`ACMAT01`, `MCMAT01`, or `PCMAT01`). If you are applying more than one growth or control matrix (i.e. `SMK_NUM_CNTLMAT` is greater than one), **Cntlmat** will open additional matrices as needed (`ACMAT[xx]`, `MCMAT[xx]`, or `PCMAT[xx]`). **Cntlmat** also uses the inventory file output from **Smkinven** (`AREA`, `MOBL`, or `PNTS`) and the country, state and county file (`COSTCY`).

Grwinven outputs new SMOKE intermediate inventory files (`AREA_O`, `MOBL_O`, or `PNTS_O`) that have been controlled and/or grown into the past or future. Note that the suffix _O is used to distinguish the **Grwinven** output files from the input files. You can also choose to output ASCII inventory files in ORL format (`ARINV_O`, `MBINV_O`, or `PTINV_O`). **Grwinven** also writes a log file (`LOGFILE`) and can create an ASCII inventory file for activity data (`MBINV_AO`).

6.5.3.1. Input Files

File Name	Format	Description
`[A|M|P]CMAT_01`	I/O API NetCDF	Control or growth matrix output from **Cntlmat**; total number of files is determined by `SMK_NUM_CTLMAT` setting
`[A|M|P]CMAT_02,` `[A|M|P]CMAT_03,` ...	I/O API NetCDF	Additional control or growth matrices; total number of files is determined by `SMK_NUM_CTLMAT` setting
`[AREA|MOBL|PNTS]`	SMOKE map	Intermediate inventory file produced by **Smkinven**
`COSTCY`	ASCII	Country, state, and county file
`GEO-CODE_LEVEL[1-4]`	ASCII	Geographical level codes and descriptions for gridded inventories, only when USE_EXP_GEO_CODES is set to Y, (COSTCY is ignored when `GEOCODE_LEVEL[1-4]` files are used)

6.5.3.2. Input Environment Variables

- `USE_EXP_GEO_CODES`: [default: N]

  If set to Y then use GEOCODE_LEVEL[1-4] files instead of the COSTCY file to define expanded geographical detail such as country-level(GEOCODE_LEVEL1), country/state-level (GEO-CODE_LEVEL2), country/state/county-level (GEOCODE_LEVEL3) or country/state/county/tribal-level (GEOCODE_LEVEL4).

- `ORL_NONROAD_OUT`: [default: N]

  - Y: Output NONROAD inventory in ORL format

  - N: Do not output NONROAD inventory in ORL format

- `SMK_GRWORLOUT_YN`: [default: N]

  Indicates if an ORL-formatted grown and/or controlled inventory should be output. Currently only ORL nonpoint ond point inventory formats are supported. `SMK_GRWORLOUT_YN` and `SMK_GRWIDAOUT_YN` cannot be both set to Y in the same run. In SMOKE version 2.3, the Grwinven ORL output format was changed to use a more compact structure (eliminated spaces), uses commas, and includes semi-colons around all text fields. These changes make these files much smaller and easier to import to other software, but unfortunately harder to read in a text editor.

  - Y: Output inventory in ORL format

  - N: Do not output inventory in ORL format

- `SMK_GRWSMKOUT_YN`: [default: Y]

  - Y: Output inventory in SMOKE I/O API format

  - N: Do not output inventory in SMOKE I/O API format

- `SMK_NUM_CTLMAT`: [default: 1]

  Number of control and/or growth matrices to import and apply to the inventory.

- `SMK_TMPDIR`: [default: .]

  Defines the path to use for temporary files that are written out when supplying the inventory.

6.5.3.3. Output Files

File Name	Format	Description
`[AREA_O|MOBL_O|PNTS_O]`	SMOKE map	Annual and average-day inventory emissions and source characteristics
`[ARINV_O|MBINV_O|PT-INV_O]`	ASCII	Annual and average-day inventory emissions and source characteristics
`LOGFILE`	ASCII	Log generated from executing **Grwinven**
`MBINV_AO`	ASCII	Annual activity inventory and source characteristics

6.6. Laypoint

6.6.1. Description

Processing categories: point

The **Laypoint** program uses the SMOKE point-source inventory file with gridded and hourly meteorology data to compute hourly plume rise for all point sources. The plume rise is expressed in terms of layer fractions for each source.

Laypoint is run only for the MAQSIP and CMAQ styles of processing, in which all of the plume rise calculations are done prior to input to the AQM. In **Laypoint**, meteorological data are used to calculate the plume rise for all point sources. Then, the plume is distributed into the vertical layers that the plume intersects based on the pressure in each layer. Only these fractions are stored (not the emissions in each layer), until the SMOKE merge step (**Smkmerge**). Therefore, the plume rise calculations do not need to be repeated for each control strategy or grid. The **Laypoint** program calculates the plume rise for all days of each meteorological scenario.

6.6.1.1. Plume rise calculation for fires

Regardless whether the heat flux for wildfires is precomputed or internally computed, the Pouliot-Godowitch plume rise algorithm (http://www.epa.gov/ttn/chief/conference/ei14/session12/pouliot.pdf) converts the heat flux (Q) to a bouyancy flux (F) for use in the plume rise calculation (Equation 1). Hourly heat flux (Q) can be estimated based upon daily area burned, fuel loading, material burned and a constant default heat content (8000 BTU/lb). See more detail information in Section 4.4.17, "Plume Rise Calculation for Fires" [167].

To account for a smoldering fraction of the fire emissions, fire plumes are distributed into both the model layers below the plume bottom and into the model layers between the plume top and bottom. The smoldering fraction of the total fire emissions is distributed into the model layer below the plume bottom. The remaining part of emissions are distributed into the model layers between the plume bottom and top.

Equation 2 estimates the plume bouyant efficiency (BE) which is the portion of heat release from a fire that produces bouyancy, using the fire size in acres. The smoldering fraction ($S_{fraction}$) of the fire plume is calculated from the BE (Equation 3). All the fire emissions are allocated to the model layers using a weighting based on the hourly sigma-pressure vertical coordinate. These equations are is used in **Laypoint** when the variable FIRE_PLUME_YN is set to Y. More information about processing fire emissions is in Section 4.4.17, "Plume Rise Calculation for Fires" [167].

Fire-specific plume rise equations are following:

1. $F = 0.00000258 \times Q$

2. $BE_{size} = 0.0703 \times \ln(\textit{acres}) + 0.3$

3. $S_{fraction} = 1 - BE_{size}$

where

* F = Bouyancy flux (m^4/s^3)

* Q = Heat flux for fires (BTU/hr)

* BE_{size} = bouyanct efficiency

- acres = fire size in acres

- $S_{fraction}$ = smoldering fration

6.6.2. Processing Order

For elevated point sources, **Smkinven** and **Elevpoint** must be run before **Laypoint** if using PinG and/or specific major point sources.

Laypoint must be run before elevated sources can be processed using **Smkmerge**.

6.6.3. Files and Environment Variables

Figure 6.6. Laypoint input and output files
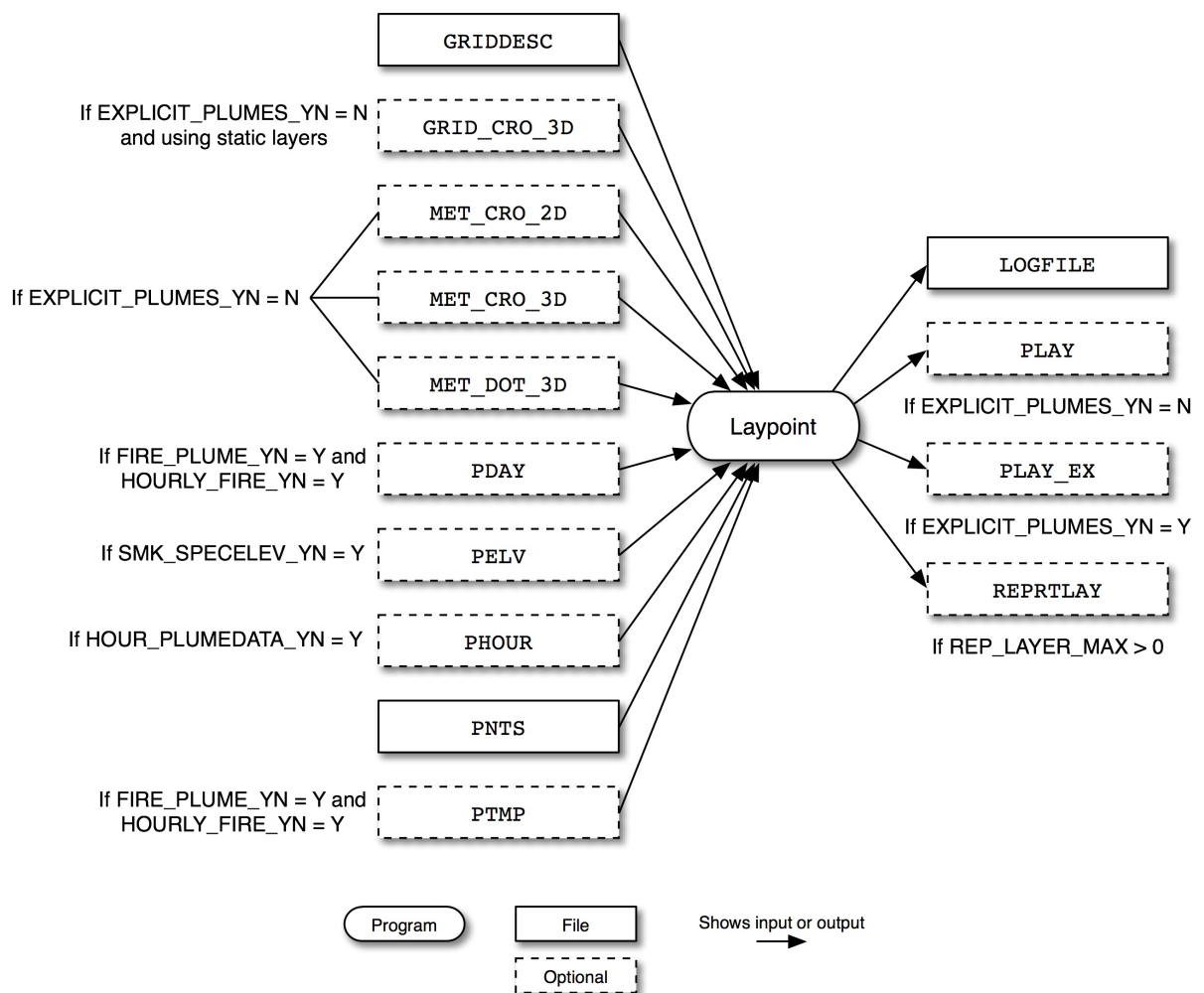
Figure 6.6, "**Laypoint** input and output files" [236] shows the input and output files for the **Laypoint** program. The input files include the grid description file (GRIDDESC) and meteorology files created by the meteorology preprocessor (GRID_CRO_3D, MET_CRO_2D, MET_CRO_3D, and MET_DOT_3D). If elevated sources have been selected, **Laypoint** uses the the elevated point source identification file output from **Elevpoint** (PELV). Hourly plume data can be read from the PHOUR hourly inventory file and annual in-

ventory data is read from the `PNTS` file, both output from **Smkinven**. When processing fire sources, the `PDAY` file created by **Smkinven** and the `PTMP` file created by **Temporal** may be used.

Laypoint outputs a log file (`LOGFILE`), hourly plume rise layer fractions for all sources (`PLAY`) or optionally only for explicit sources (`PLAY_EX`), and a report file `REPRTLAY` if reporting is turned on using the `REP_LAYER_MAX` setting.

6.6.3.1. Input Files

File Name	Format	Description
`GRIDDESC`	ASCII	Description of the modeling grid
`GRID_CRO_3D` (optional)	I/O API NetCDF	Meteorology file produced by the meteorology preprocessor; only needed if using static layer information
`MET_CRO_2D`	I/O API NetCDF	Meteorology file produced by the meteorology preprocessor
`MET_CRO_3D`	I/O API NetCDF	Meteorology file produced by the meteorology preprocessor
`MET_DOT_3D`	I/O API NetCDF	Meteorology file produced by the meteorology preprocessor
`PDAY` (optional)	I/O API NetCDF	Daily point-source inventory file produced by **Smkinven**
`PELV` (optional)	ASCII	Elevated point source identification file produced by **Elevpoint**
`PHOUR` (optional)	I/O API NetCDF	Hourly point-source inventory file produced by **Smkinven**
`PNTS`	SMOKE map	Intermediate inventory file produced by **Smkinven**
`PTMP`	I/O API NetCDF	Hourly emissions file produced by **Temporal**

6.6.3.2. Input Environment Variables

- `EXPLICIT_PLUMES_YN`: [default: N]

  Indicates whether only explicit plume rise sources should be calculated.

  - Y: Process only explicit plume rise sources; automatically sets `HOUR_PLUMEDATA_YN` to Y

  - N: Process all sources

- `FIRE_AREA`: [default: 0.0]

  Sets daily area burned in acres/day; only used when `FIRE_PLUME_YN` is set to Y and `HOURLY_FIRE_YN` is set to N.

- `FIRE_HFLUX`: [default: 0.0]

  Sets hourly heat flux value in BTU/hr; only used when `FIRE_PLUME_YN` is set to Y and `HOURLY_FIRE_YN` is set to N.

- `FIRE_PLUME_YN`: [default: N]

  Indicates if fire-specific plume rise calculations should be used.

  - Y: Use fire-specific plume rise calculations for all sources

  - N: Use traditional plume rise calculations based on stack parameters for all sources

- `HOUR_PLUMEDATA_YN`: [default: N]

Indicates whether hourly plume rise data should be used.

- Y: Use hourly data; requires PHOUR file

- N: Do not use hourly data

- HOURLY_FIRE_YN: [default: N]

  Indicates if hourly fire data should be used; only checked when FIRE_PLUME_YN is set to Y.

  - Y: Read hourly heat flux from PTMP file and daily area burned from PDAY file

  - N: Get heat flux and area burned from environment variables FIRE_AREA and FIRE_HFLUX

- IOAPI_ISPH: See description in Section 6.3.3.2, "Input Environment Variables" [226]

- PLUME_GTEMP_NAME: [default: TEMP2]

  Sets ground temperature variable name from meteorology file for plume rise calcuation.

- REP_LAYER_MAX: [default: -1]

  Layer number for reporting high plume rise; reporting is turned off if set to less than 1.

- SMK_EMLAYS: [default: -1]

  Number of emissions layers; must be at least 4 otherwise program will not run.

- SMK_SOURCE: See description in Section 6.2.3.2, "Input Environment Variables" [221].

- SMK_SPECELEV_YN: [default: N]

  Defines whether the **Elevpoint** output file PELV will be used to set elevated sources.

  - Y: Use the PELV file to set elevated sources during run

  - N: Let **Laypoint** plume rise calculation determine whether or not sources are elevated

- USE_EDMS_DATA_YN: [default: N]

  Process EDMS aviation source elevation height as its plume rise and distributes the emissions into the layer where it belongs to.

  - Y: Skip plume rise calculation and distribute EDMS aviation emissions into the layer where it belongs to using its elevation height.

  - N: Do not process EDMS aviation sources

- VELOC_RECALC: [default: N]

  Determines whether velocity should be recalculated based on hourly flow rates and stack diameter.

  - Y: Recalculate velocity for hourly data

  - N: Use velocity values from hourly inventory file

- VERTICAL_SPREAD: [default: 0]

Sets the vertical spread method for determining the bottom and top heights of each plume.

- 0: Use the default Turner approach where plume thickness equals the amount of plume rise

- 1: Compute vertical spread using ambient temperature data

6.6.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Laypoint**
PLAY	I/O API NetCDF	Hourly plume rise layer fractions by source
PLAY_EX	I/O API NetCDF	Hourly plume rise layer fractions for explicit sources
REPRTLAY	ASCII	Report file for plumes exceeding specified layer

6.7. Met4moves

6.7.1. Description

Processing categories: mobile

The **Met4moves** is a meteorological processor that prepares temperature and relative humidity (RH) data for use by both MOVES and SMOKE. **Met4moves** produces specific meteorological metrices for the reference county(s) for MOVES and additional meteorological metrics for all inventory counties in the county group for SMOKE

6.7.1.1. Definition of a "day"

Met4moves seeks to obtain 24 hours' worth of meteorology data for a source for each day, which is necessary for generating emission factors. However, the definition of a "day" differs for sources in different time zones. By default, **Met4moves** uses each source's time zone to determine a day for that time zone. To instead use the same day-start-time for all sources, the UNIFORM_STIME environment variable can be used.

If the inventory covers multiple time zones, all sources will not have the same coverage of days. **Met4moves** will use meteorology data from previous or subsequent days to fill in missing data. If appropriate data are not available, the program will quit with an error and you must provide additional meteorology data to cover the time period of interest. Currently, **Met4moves** does not considers Daylight Saving time in its determination of "day".

6.7.2. Processing Order

Met4moves must be run prior to MOVES processing scripts. The SMOKE Utility program, **Metcombine**, needs to be run prior to **Met4moves**. **Metcombine** allows the user to combine ambient temperature, pressure and mixing ratio variables from two different MCIP files (METCRO_2D and METCRO_3D) that are needed to compute Relative Humidity (RH) in **Met4moves**.

6.7.3. Files, Environment Variables and Run Syntax

Figure 6.7. Met4moves input and output files
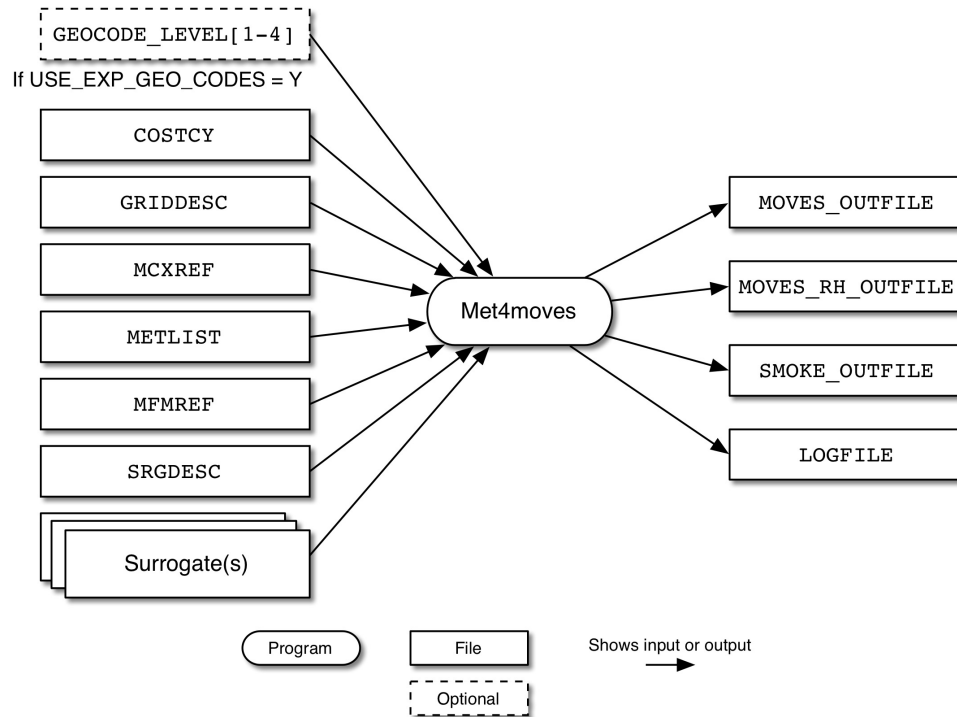
Figure 6.7, "**Met4moves** input and output files" [240] shows the input and output files for the **Met4moves** program. The input files are the Grid Description File from MCIP(`GRIDDESC`), the Surrogate Description file (`SRGDESC`), the MCIP List of Meteorology File (`METLIST`), and the reference county cross-reference file (`MCXREF`).

Met4moves outputs datasets for SMOKE that contain the daily gridded min/max temperatures and averaged RH. **Met4moves** outputs for MOVES datasets that provide all (min/max) temperature combinations for a reference county, reflecting all min/max combinations for all inventory counties in that county group. The associated RH values are also included in these datasets. In addition, the datasets include the 24-h diurnal profiles needed for the RPP emission process, and contain user-specified temperature increments for use by MOVES.

6.7.3.1. Input Files

File Name	Format	Description
COSTCY	ASCII	Country, state, county codes file, with time zones
GEO-CODE_LEVEL[1-4]	ASCII	Geographical level codes and descriptions for gridded inventories, only when USE_EXP_GEO_CODES is set to Y, (COSTCY is ignored when GEOCODE_LEVEL[1-4] files are used)
GRIDDESC	ASCII	Modeling Domain Grid Description File

File Name	Format	Description
MCXREF	ASCII	Reference county cross-reference file; NOTE: The list of counties in the MCXREF file used by **Met4moves** must match the list of inventory counties used by SMOKE
METLIST	ASCII	A list of meteorology files containing temperature, mixing ratio, and surface pressure data created by **Metcombine**
MFMREF	ASCII	Reference county fuel monthcross-reference file; NOTE: The list of counties in the MFMREF file used by **Met4moves** must match the list of inventory counties used by SMOKE
SRGDESC	ASCII	Surrogate Description File
Surrogate File(s)	ASCII	Surrogate file(s) selected by SRG_LIST

6.7.3.2. Input Environment Variables

- STDATE: [default: 0]

  This environment variable sets the overall episode start date: Julian format (YYYYDDD) for **Movesmrg**.

- ENDDATE: [default: 0]

  Sets the overall episode end date; Julian format (YYYYDDD).

- ENDTIME: [default: 230000]

- USE_EXP_GEO_CODES: [default: N]

  If set to Y then use GEOCODE_LEVEL[1-4] files instead of the COSTCY file to define expanded geographical detail such as country-level(GEOCODE_LEVEL1), country/state-level (GEO-CODE_LEVEL2), country/state/county-level (GEOCODE_LEVEL3) or country/state/county/tribal-level (GEOCODE_LEVEL4).

- PD_TEMP_INCREMENT: [default: 5]

  Defines the temperature increment (in deg. F) for RPD lookup table to create combinations of min/max temperature bins

- PH_TEMP_INCREMENT: [default: 5]

  Defines the temperature increment (in deg. F) for RPH lookup table to create combinations of min/max temperature bins

- PV_TEMP_INCREMENT: [default: 5]

  Defines the temperature increment (in deg. F) for RPV lookup table to create combinations of min/max temperature bins

- PP_TEMP_INCREMENT: [default: 10]

  Defines the temperature increment (in deg. F) for RPP lookup table to create combinations of min/max temperature bins

- MIN_NO_RH_BY_TEMPBIN: [default: 1]

Defines the minimum number of data point for computing averaged RH by tempearature bin

- `SPECIFIC_HUMIDITY_YN`: [default: N]

  Defines whether compute specific humidity instead of relative humidity

- `SRG_LIST`: [default: N]

  Specifies the code(s) of the spatial surrogate(s) to be used in selecting the grid cells for the county (example: setenv SRG_LIST '100,230').

- `SRGPRO_PATH`: [default: N]

  Defines the location of spatial surrogate files.

- `TEMP_BUFFER_BIN`: [default: 0.0]

  Defines the temperature buffer value (°F). This value will allow **Met4moves** to expand the min/max temperature range for MOVES RPP processing. If `TEMP_BUFFER_BIN` is set to 10.0, the original max temperature which is 90.0°F will be treated as 100.0°F, and minimum temperaure (=30°F) will be treated as 20.0°F internally in **Met4moves**.

- `TVARNAME`: [default: TEMP2]

  Specifies the variable name for the temperature to extract from the **Met4moves** output file (`MOVES_OUTFILE`).

6.7.3.3. Output Files

File Name	Format	Description	
MOVES_OUTFILE	ASCII	Contains the absolute minimum and maximum temperatures and average RH values and the 24 hour temperature profiles associated with each reference county (example filename: MOVES_12km_2009091-2009120.txt)	
MOVES_RH_OUT-FILE	ASCII	Contains the reference county-specific average RH, minimum and maximum temperature by temperature bin and fuel month.	
SMOKE_OUTFILE	ASCII	Contains gridded daily min/max temperatures in local timei for RPP mode run in **Movesmrg**	
LOGFILE	ASCII	Log generated from executing **Met4moves**	

6.8. Movesmrg

6.8.1. Description

Processing categories: mobile

Movesmrg is a new, mobile-sources-only program, loosely based on the SMOKE program **Smkmerge**, that combines the intermediate files produced by the other SMOKE programs (i.e., **SmkInven**, **Grdmat**, **Spcmat**, **Temporal**). The **movesmrg** program is discussed in detail in Section 6.8, "**Movesmrg**" [242]. Similar to **Smkmerge**, the main goal of **Movesmrg** is to create (1) gridded, speciated, hourly air quality

model-ready input files and (2) county-level SCC summaries. The primary difference between **Movesmrg** and **Smkmerge** is that **Movesmrg** uses the SMOKE Formatted MOVES emission rate lookup tables to compute emissions on the fly, by multiplying the emissions factors by hourly VMT (for RatePerDistance [RPD]), annual or monthly hotelling hours (for RatePerHour [RPH]) or by monthly or annual vehicle populations (for RatePerVehicle [RPV] or RatePerProfile [RPP]).

Movesmrg must be run four times to compute on-roadway and off-network (including vapor venting) emissions processes individually, since the emission rate calculation methods in SMOKE for these categories are quite different. Once all three emissions processes have been computed, the emission output files can be merged (optionally along with other 2D sectors separate from onroad) using **Mrggrid**, which combines any number of 2D or 3D hourly, gridded, and speciated emission data files. See the section Section 6.10, "**Mrggrid**" [251] for more details on the above programs.

- **RPD, RPH and RPV Emission Calculations**

  For its RPD, RPH and RPV lookup table calculations, **Movesmrg** uses the gridded hourly temperatures in GMT, and then adjusts the hour of the RPD, RPH and RPV lookup tables from local time to GMT to get the emission rates for the reference county for the appropriate month. The key lookup fields for the factors are hourly temperature and avarage speed for the RPD lookup table calculation, hourly temperature for the RPH lookup table calculation, and hourly temperature and hour of the day in local time for the RPV table calculation.

- **RPP Emission Calculations**

  For the RPP lookup table calculations, **Movesmrg** uses the RPP lookup table as an input file for the emission rates that depend not on the gridded hourly temperatures, but on the temperature profiles output by the **Met4moves** program. The emissions estimates based on the RPP table are computed for the temporal averaging period specified for the temperature profiles, rather than using hourly gridded meteorology.

6.8.2. Processing Order

Movesmrg must be run after **Met4moves** and MOVES and before **Mrggrid**.

6.8.3. Files and Environment Variables

Figure 6.8. Movesmrg input and output files
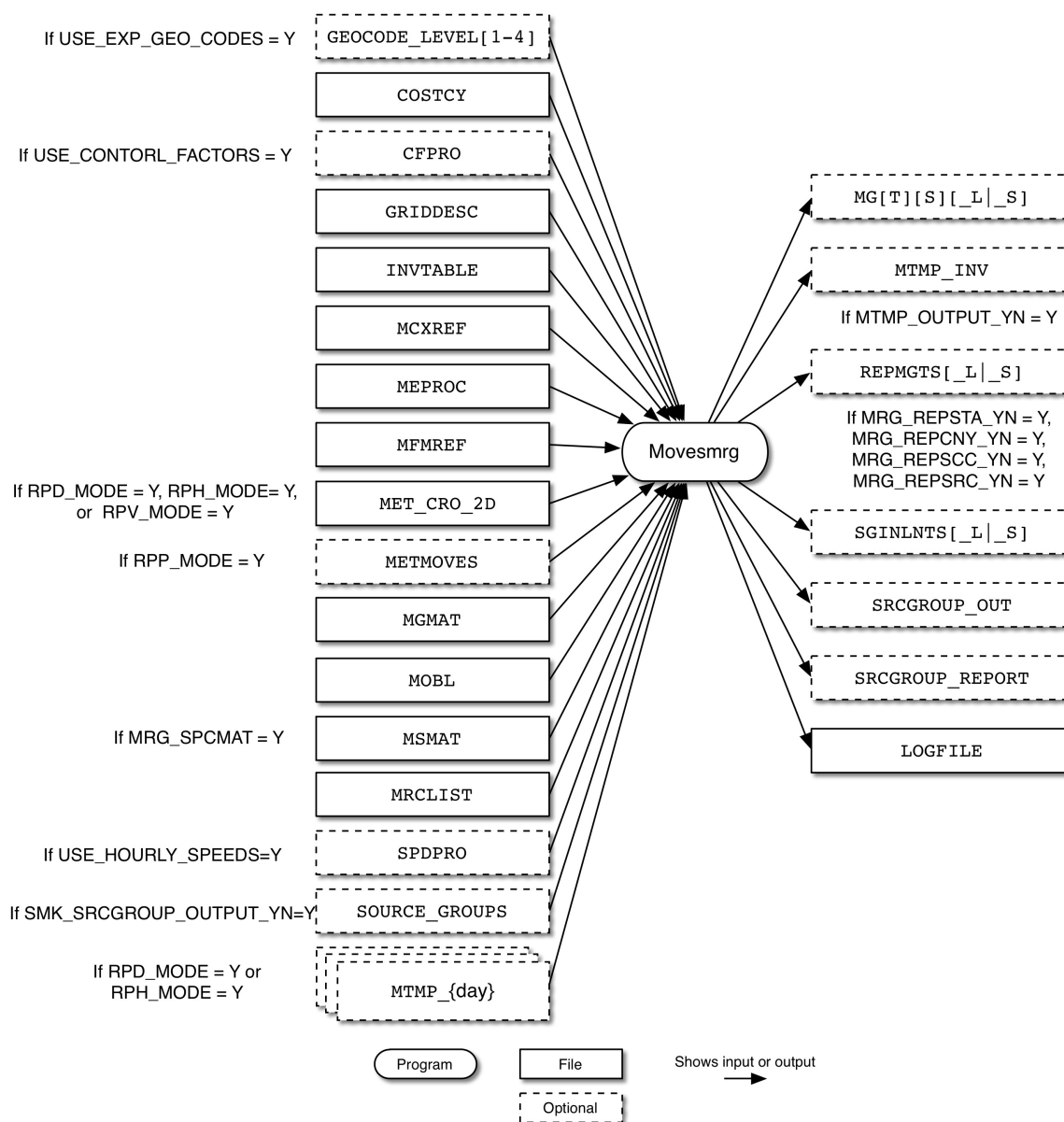
Figure 6.8, "**Movesmrg** input and output files" [244] shows the input and output files for the **Movesmrg** program. The input files are the country, state, and county codes file (`COSTCY`), the grid description file (`GRIDDESC`), the inventory data table (`INVTABLE`), the reference county fuel month file (`MFMREF`), the list of emission process and associated pollutants file (`MEPROC`), the reference county cross-reference file (`MCXREF`), the list of emission factors files by reference county file (`MRCLIST`), the meteorology files (`MET_CRO_2D`), the gridding matrix from **Grdmat** (`MGMAT`), the inventory file output from **Smkinven** (`MOBL`), the speciation matrix from **Spcmat** (`MSMAT_L` or `MSMAT_S`), the **Met4moves** output file for SMOKE (`METMOVES =SMOKE_OUTFILE`), the MOVES-based lookup tables (RPD, RPV, and RPP), and the daily hour VMT files from **Temporal** (`MTMP_(day)`). Note that `MET_CRO_2D`, `METMOVES`, and `MTMP_(day)` are optional files that are included only in certain situations.

Movesmrg outputs a model-ready emissions file `MGTS_L` or `MGTS_S`, a report file `REPMGTS_S`, and a log file `LOGFILE`.

When using source apportionment (see Section 4.4.23, "Output source apportionment data" [175]), **Movesmrg** will also read the source grouping file (`SOURCE_GROUPS`) and output the source group information `SRCGROUPS_OUT`, the emissions for each source group `SGINLNTS_L` or `SGINLNTS_S`, and a source group report file `SRCGRP_REPORT`.

6.8.3.1. Input Files

File Name	Format	Description
COSTCY	ASCII	Country, state and county codes file
GEO-CODE_LEVEL[1-4]	ASCII	Geographical level codes and descriptions for gridded inventories, only when USE_EXP_GEO_CODES is set to Y, (COSTCY is ignored when GEOCODE_LEVEL[1-4] files are used)
CFPRO (optional)	ASCII	Control factor file
GRIDDESC	ASCII	Grid description file
INVTABLE	ASCII	Inventory data table
MFMREF	ASCII	Reference county fuel month file
MEPROC	ASCII	List of emission process and associated pollutants file
MCXREF	ASCII	Reference county cross-reference file
MRCLIST	ASCII	List of emission factors files (i.e., RPD, RPV and RPP Lookup tables) by reference county file
METLIST	ASCII	List of MCIP meteorology files
MGMAT	ASCII	Gridding matrix from **Grdmat**
MOBL	ASCII	Inventory file output from **Smkinven**
MSMAT_[L|S]	ASCII	Speciation matrix from **Spcmat**
METMOVES (=SMOKE_OUT-FILE) (optional)	ASCII	Meteorology file from **Met4moves**
MTMP_{day}	ASCII	Daily hour VMT files from **Temporal**
SOURCE_GROUPS (optional)	ASCII	Source apportionment groups
SPDPRO (optional)	ASCII	Weekday and weekend hourly speed profiles

6.8.3.2. Input Environment Variables

- `USE_EXP_GEO_CODES`: [default: N]

  If set to Y then use GEOCODE_LEVEL[1-4] files instead of the COSTCY file to define expanded geographical detail such as country-level(GEOCODE_LEVEL1), country/state-level (GEOCODE_LEVEL2), country/state/county-level (GEOCODE_LEVEL3) or country/state/county/tribal-level (GEOCODE_LEVEL4).

- `MEMORY_OPTIMIZE_YN: ` : [default: Y]

Allows **Movesmrg** to use less RAM memory. However, it will increase computational time of RPD_MODE and RPV_MODE runs.

- Y: Runs **Movesmrg** in memory optimization mode.

- N: Runs **Movesmrg** faster mode using high capacity of RAM memory.

- MRG_BYDAY: [default: blank]

  String code that specifies which source category is to use a by-day approach for the hourly input files. It must be set to "M" for a **Movesmrg** to create an output file for each day.

- MRG_GRDOUT_YN: [default: N]

  Determines whether **Movesmrg** produces a gridded output file.

  - Y: Read the gridding matrix produced by the **Grdmat**

  - N: Do not read the gridding matrix.

- MRG_SPCOUT_YN: [default: N]

  Determines whether **Movesmrg** produces a speciated output file.

  - Y: Read the gridding matrix produced by the **Spcmat**

  - N: Do not read the gridding matrix.

- MRG_REPCNY_YN: [default: N]

  Determines whether **Movesmrg** produces a report of emission totals by county

  - Y: Produce the report by **Movesmrg**

  - N: Do not produce the report

- MRG_REPSTA_YN: [default: N]

  Determines whether **Movesmrg** produces a report of emission totals by state

  - Y: Produce the report by **Movesmrg**

  - N: Do not produce the report

- MRG_REPSCC_YN: [default: N]

  Determines whether **Movesmrg** produces a report of emission totals by SCC

  - Y: Produce the report by **Movesmrg**

  - N: Do not produce the report

- MRG_REPSRC_YN: [default: N]

  Determines whether **Movesmrg** produces a report of emission totals by source.

  - Y: Produce the report by **Movesmrg**. It will reset MRG_REPCNY_YN and MRG_REPSCC_YN to N to avoid to outputing duplicate information into the report.

- N: Do not produce the report

- If `MRG_GRDOUT_UNIT` [default: units]

  String code that permits user to change the unit of the gridded output file.

- If `MRG_TOTOUT_UNIT` [default: units]

  String code that permits user to change the unit of the state/county total report file.

- `MTMP_OUTPUT_YN:` [default: N]

  Defines whether **Temporal** outputs the **Temporal** intermediate output file from Section 9.14, "**Temporal**" [480] for **Smkreport** to create hourly emissions summary report based on hourly gridded emissions estimated by **Movesmrg**.

  - Y: Output Temporal intermediate output file

  - N: Do not output Temporal intermediate output file

- `RPD_MODE:` [default: N]

  Determines whether **Movesmrg** processes rate-per-distance emissions.

  - Y: Process rate-per-distance emissions

  - N: Do not process rate-per-distance emissions

- `RPH_MODE:` [default: N]

  Determines whether **Movesmrg** processes rate-per-hour emissions.

  - Y: Process rate-per-hour emissions

  - N: Do not process rate-per-hour emissions

- `RPV_MODE:` [default: N]

  Determines whether **Movesmrg** processes rate-per-vehicle emissions.

  - Y: Process rate-per-vehicle emissions

  - N: Do not process rate-per-vehicle emissions

- `RPP_MODE:` [default: N]

  Determines whether **Movesmrg** processes rate-per-profile emissions.

  - Y: Process rate-per-profile emissions

  - N: Do not process rate-per-profile emissions

- `SMK_MVSPATH:` [default: blank]

  Defines the location of MOVES emission factors lookup tables.

- `SMK_SOURCE:` [default: blank]

Sets the types of source category. It must be set to "M" for **Movesmrg**.

- `SMK_SRCGROUP_OUTPUT_YN`: [default: N]

  Indicates if source apportionment emissions should be output

  - Y: Group sources and output source apportionment files

  - N: Do not perform any source apportionment processing

- `TEMP_BUFFER_BIN`: [default: 0.0]

  Defines the temperature buffer value (°F). This value allows **Movesmrg** to cover the outside of min/max temperature range in `SMOKE_OUTFILE` from **Met4moves** for **Movesmrg** RPP processing. If `TEMP_BUFFER_BIN` is set to 10.0, the original max temperature which is 90.0°F will be treated as 100.0°F, and minimum temperaure (=30°F) will be treated as 20.0°F internally in **Movesmrg**. Emission rates for those temperatures outside of original temperature range will share the ones from original max/min temperatures.

- `TMPVNAME`: [default: TEMP2]

  Name of temperature variable to read from the meteorology files for **Movesmrg**.

- `USE_CONTROL_FACTORS`: [default: N]

  Allows the user to apply control factor by region, SCC, pollutant, and month.

  - Y: Apply control factors to MOVES emission rates from MOVES lookup tables.

  - N: Do not apply control factor.

- `USE_EXP_CONTROL_FAC_YN`: [default: N]

  Allows the user to optionally apply pollutant/species-specific explict control factor by region, SCC, and month when `USE_CONTROL_FACTORS` is set to Y.

  - Y: Apply pollutant/species-specific explicit control factors to MOVES emission rates from MOVES lookup tables.

  - N: Do not apply control factor.

- `USE_REF_CONTROL_FAC_YN`: [default: N]

  Allows the user to optionally apply reference-county-level control factor by SCC, and month when `USE_CONTROL_FACTORS` is set to Y.

  - Y: Apply reference-county-level control factors to MOVES emission rates from MOVES lookup tables.

  - N: Do not apply control factor.

- `USE_HOURLY_SPEEDS`: [default: N]

  Determine whether processes weekday/weekend hourly speed profiles or not.

  - Y: Process weekday/weekend hourly speed profiles

- N: Do not process hourly speed profiles

- `USE_SPCMAT_SPC_YN`: [default: N]

  Determine whether estimate air quality model species using chemical speciation profiles or not.

  - Y: Estimate model species emissions using GSPRO input file

  - N: Estimate model species emissions directly from MOVES emission rates lookup tables

6.8.3.3. Output Files

File Name	Format	Description
`MGTS[_S|_L]`	I/O API NetCDF	Model Ready Emissions File
`MTMP_INV`	I/O API NetCDF	Hourly emissions by source
`REP-MGTS[_S|_L]`	ASCII	Report File
`LOGFILE`	ASCII	Logfile
`SGIN-LNTS[_S|_L]`	I/O API NetCDF	Emissions by source group, formatted as CMAQ in-line emissions data, produced if `SMK_SRCGROUP_OUTPUT_YN` is set to Y
`SRCGROUPS_OUT`	I/O API NetCDF	Source group information, produced if `SMK_SRCGROUP_OUTPUT_YN` is set to Y
`SRCGRP_REPORT`	ASCII	Source group report file, produced if `SMK_SRCGROUP_OUTPUT_YN` is set to Y

6.9. Mrgelev

6.9.1. Description

Processing categories: point

The **Mrgelev** merges multiple ASCII elevated files created by **Smkmerge**. The program can also output a binary elevated file needed by CAMx, REMSAD, and UAM models. **Mrgelev** also handles matching PinG sources from different inputs, ensuring that only the top 300 emission sources are marked as PinG sources in the final merged output. For PinG matching, **Mrgelev** uses `PELV` files output by **Elevpoint**. **Mrgelev** uses the environment variable `MRG_DIFF_DAYS` to merge files from different days into a single file.

6.9.2. Processing Order

Mrgelev is used to merge ASCII elevated files created by **Smkmerge**. If PinG matching is desired, then output files from **Elevpoint** are required.

The output binary elevated file from **Mrgelev** is used when running CAMx, REMSAD, or UAM models.

6.9.3. Files and Environment Variables

Figure 6.9. Mrgelev input and output files
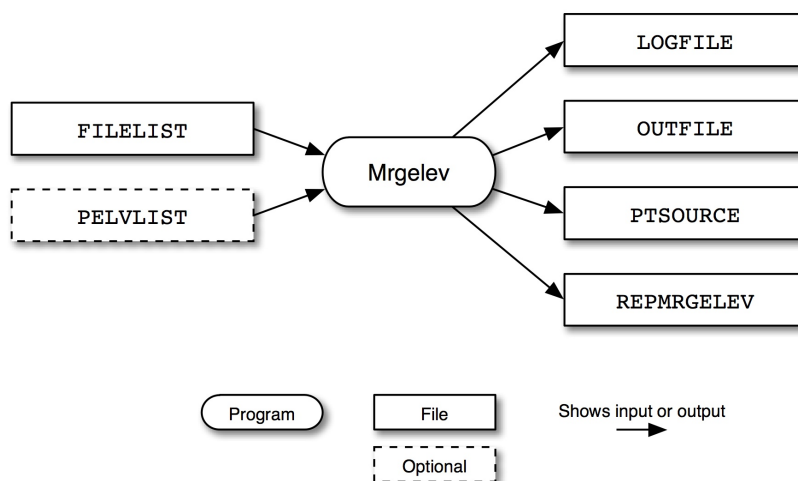
Figure 6.9, "**Mrgelev** input and output files" [250] shows the input and output files for the **Mrgelev** program. The input files are a list of the ASCII elevated files (`FILELIST`) and optionally a list of `PELV` files created by **Elevpoint** (`PELVLIST`). In both files, the complete file name including path must be provided. When using the `PELVLIST` file, each file must match the corresponding ASCII elevated file. The program outputs a log file (`LOGFILE`), a merged ASCII elevated file (`OUTFILE`), a merged binary elevated file (`PT-SOURCE`), and a report file (`REPMRGELEV`).

6.9.3.1. Input Files

File Name	Format	Description
FILELIST	ASCII	List of input ASCII elevated files including path and file name
PELVLIST	ASCII	List of input PELV files including path and file name; leave environment variable unset if not processing PinG sources

6.9.3.2. Input Environment Variables

- `MRG_DIFF_DAYS`: [default: N]

  This environment variable allows **Mrgelev** to merge files with data from different days.

  - Y: Allow data from different dates to be merged together

    Mrgelev will use the environment variables `G_STDATE`, `G_STTIME`, `G_RUNLEN`, and `G_TSTEP` to set the start date, start time, length, and time step of the output file. All input files must have the same starting time as `G_STTIME`, the same time step as `G_TSTEP`, and must contain more than `G_RUNLEN` steps of data. For each input file, if it contains data to cover the actual episode requested by `G_STDATE`, `G_STTIME`, and `G_RUNLEN`, then the data for the actual episode will be used; otherwise, data starting at the first time step of the file will be used.

  - N: Require all merged data to be from the same date

Mrgelev will determine the output file time period based on the latest starting input file and the earliest ending input file.

6.9.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Mrgelev**
OUTFILE	ASCII	Merged ASCII elevated file
PTSOURCE	Binary	Merged binary elevated file
REPMRGELEV	ASCII	Report file summarizing input and output emissions for QA purposes

6.10. Mrggrid

6.10.1. Description

Processing categories: all

The purpose of the **Mrggrid** program is to combine gridded emission data files, which can be speciated or nonspeciated, and hourly or time-independent. It can combine any number of 3-D emissions files with any number of 2-D emissions files. In fact, this program will work with any I/O API NetCDF gridded files and need not be used only for emissions files. The program merges whatever data variables are in the input files. These files can each have different sets of data variables; **Mrggrid** will sum values only for each variable that is common to two or more files.

Mrggrid checks several features of the input files to ensure that the data in these files are appropriate to be merged together. All input files must have use the same time step. **Mrggrid** compares the units of the variables with the same name and it creates an error message when the units are not identical. It is not sufficient to have units that are the same in meaning but different in syntax. For example, although tons/hour and tons/hr represent the same units, **Mrggrid** will not recognize this because of the different syntax. The user must resolve unit discrepancies before running the program. In addition, **Mrggrid** compares the horizontal grid settings across all 2-D and 3-D files, and these settings must be identical in all respects (e.g., number of cells, grid cell size, grid projection). Finally, **Mrggrid** checks the layer structures across multiple 3-D files. The 3-D files can have a different number of layers, as long as any layer common to two or more files has the same thickness in each of the files; **Mrggrid** also checks the I/O API type of vertical structures and top height, and these must be identical across all 3-D files.

6.10.2. Processing Order

Mrggrid can be used with any gridded output files from **Smkmerge**. It can also be used with other I/O API NetCDF gridded files produced outside of SMOKE.

Output files from **Mrggrid** are usually moodel-ready input files that will be used in running an AQM.

6.10.3. Files and Environment Variables

Figure 6.10. Mrggrid input and output files
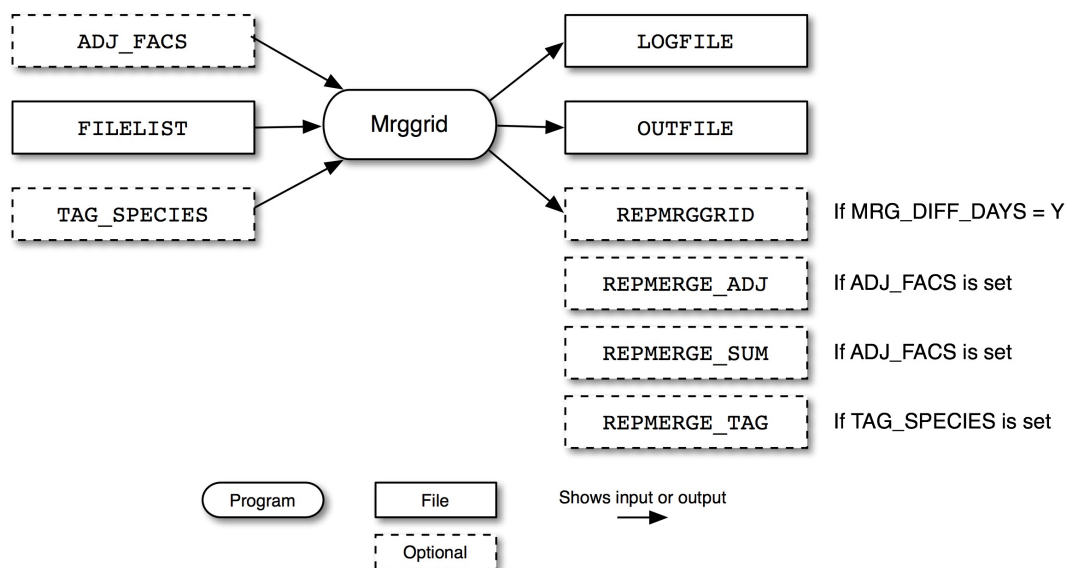
Figure 6.10, "**Mrggrid** input and output files" [252] shows the input and output files for the **Mrggrid** program. The input file is simply a list of the logical file names for the files to be merged (`FILELIST`). The program outputs a log file (`LOGFILE`), a gridded, optionally layered, optionally hourly emissions file (`OUTFILE`), and if merging different days, a report file (`REPMRGGRID`).

6.10.3.1. Input Files

File Name	Format	Description
FILELIST	ASCII	List of *logical* file name for 2-D and 3-D gridded I/O API NetCDF files; this file cannot contain the full paths and file names. Users should make sure that the logical file names that appear in `FILELIST` are defined prior to running **Mrggrid**.

ADJ_FACS
 	
(optional)
	ASCII	File that lists the species name, the logical file name, and the factor to be applied to all grid cells. The species name must be in columns A, the logical file name in column B, and the factor in columns C. SMOKE will ignore the case of the text, treating lower- and uppercase characters as if they were the same. For example, the following `ADJ_FACS` file sample would multiply NO emissions from the MGTS_L file by 1.3 (30% increase) and multiply PAR from the PGTS_L file by 0.6 (40% decrease).
		
		`NO, MGTS_L, 1.3`
		`PAR, PGTS_L, 0.6`

TAG_SPECIES (optional)	ASCII	File that lists the logical file name, the species name, and the tag name to be appended. The logical file name in column B, the species name must be in columns B, and the factor in columns

File Name	Format	Description

		C. SMOKE will ignore the case of the text, treating lower- and uppercase characters as if they were the same. For example, the following `TAG_SPECIES` file sample would append a tag (t1) to the NO species from the MGTS_L file and append a tag (t3) to the PAR from the PGTS_L file.
		
		`MGTS_L, NO,  t1`
		`PGTS_L, PAR, t3`

6.10.3.2. Input Environment Variables

- `MRG_DIFF_DAYS`: [default: N]

  This environment variable allows **Mrggrid** to merge files with data from different days.

  - Y: Allow data from different dates to be merged together

    Mrggrid will use the environment variables `G_STDATE`, `G_STTIME`, `G_RUNLEN`, and `G_TSTEP` to set the start date, start time, length, and time step of the output file. All input files must have the same starting time as `G_STTIME`, the same time step as `G_TSTEP`, and must contain more than `G_RUNLEN` steps of data. For each input file, if it contains data to cover the actual episode requested by `G_STDATE`, `G_STTIME`, and `G_RUNLEN`, then the data for the actual episode will be used; otherwise, data starting at the first time step of the file will be used.

  - N: Require all merged data to be from the same date

    Mrggrid will determine the output file time period based on the latest starting input file and the earliest ending input file.

6.10.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Mrggrid**
OUTFILE	I/O API NetCDF	Gridded, optionally layered, optionally hourly emission file
REPMRGGRID	ASCII	Report file generated when `MRG_DIFF_DAYS` is set to Y; the report lists the date and time for each output time step along with the actual date and time of the data read from each file
REPMERG_ADJ	ASCII	Report file generated when `ADJ_FACS` is set; the report lists the date, the logical file name, the species name, the factor applied, a daily total of species before and after applying adjustment factor, and the ratio (after/ratio). The report only lists when the ratio is not equal to 1.0
REPMERGE_SUM	ASCII	Report file generated when `ADJ_FACS` is set; the report lists the date, the species name, a daily total of species over all the logical files before and after applying adjustment factors, and the ratio (after/ratio). The report only lists when the ratio is not equal to 1.0

File Name	Format	Description
REPMERGE_TAG	ASCII	Report file generated when `TAG_SPECIES` is set; the report lists the logical files name, species name before and after appending the tag

6.11. Mrgpt

6.11.1. Description

Processing categories: Point

The purpose of the **Mrgpt** program is to combine inline-mode model-ready multiple point speciated hourly emission data files (not gridded) created by **Smkmerge** and ASCII elevated files created by **Elevpoint**. The program merges whatever data variables are in the input files. These files can each have different sets of data variables; **Mrgpt** will sum values only for each variable that is common to two or more files.

Mrgpt checks several features of the input files to ensure that the data in these files are appropriate to be merged together. All input files must have use the same time step. **Mrgpt** compares the units of the variables with the same name and it creates an error message when the units are not identical. It is not sufficient to have units that are the same in meaning but different in syntax. For example, although tons/hour and tons/hr represent the same units, **Mrggrid** will not recognize this because of the different syntax. The user must resolve unit discrepancies before running the program.

6.11.2. Processing Order

Mrgpt can be used with any hourly speciated emissions from **Smkmerge** and any ASCII elevated output files from **Elevpoint**.

Output files from **Mrgpt** are inline-model model-ready input files that will be used in running an AQM.

6.11.3. Files and Environment Variables

Figure 6.11. Mrgpt input and output files
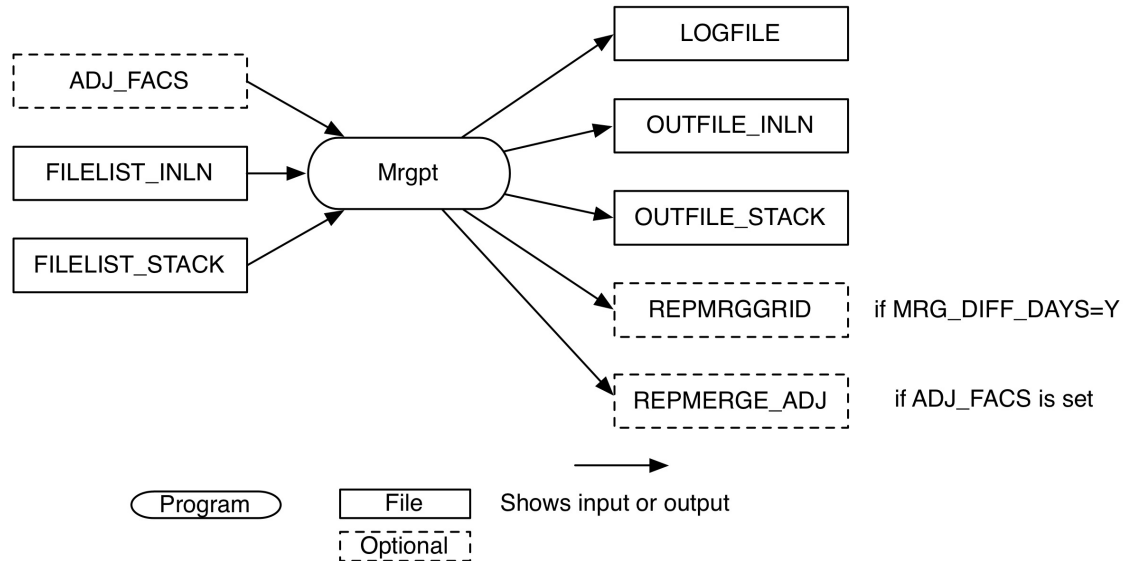
Figure 6.11, "**Mrgpt** input and output files" [255] shows the input and output files for the **Mrgpt** program. The input files are a list of the logical file names for the speciated hourly emissions files from **Smkmerge** to be merged (`FILELIST_INLN`) and a list of the logical file names for the ASCII elevated output files from **Elevpoint** to be merged (`FILELIST_STACK`). The program outputs a log file (`LOGFILE`), a merged speciated hourly emissions file (`OUTFILE_INLN`), a merged ASCII elevated output file (`OUT-FILE_STACK`), and if merging different days, a report file (`REPMRGGRID`).

6.11.3.1. Input Files

File Name	Format	Description
`FILELIST_INLN`	ASCII	List of *logical* file name for hourly speciated I/O API NetCDF files; this file cannot contain the full paths and file names. Users should make sure that the logical file names that appear in `FILELIST_INLN` are defined prior to running **Mrgpt**.
`FILEL-IST_STACK`	ASCII	List of *logical* file name for ASCII elevated output files; this file cannot contain the full paths and file names. Users should make sure that the logical file names that appear in `FILELIST_STACK` are defined prior to running **Mrgpt**.
`ADJ_FACS` (optional)	ASCII	File that lists the species name, the logical file name, and the factor to be applied to all grid cells. The species name must be in columns A, the logical file name in column B, and the factor in columns C. SMOKE will ignore the case of the text, treating lower- and uppercase characters as if they were the same. For example, the following `ADJ_FACS` file sample would multiply NO emissions from the PTIPMTS_L_INLN file by 1.3 (30% in-

File Name	Format	Description

		crease) and multiply PAR from the PTNONIPMTS_L_INLN file by 0.6 (40% decrease).
		
		`NO, PTNONIPMTS_L_INLN, 1.3`
		`PAR, PTIPMTS_L_INLN, 0.6`

6.11.3.2. Input Environment Variables

- `MRG_DIFF_DAYS`: [default: N]

  This environment variable allows **Mrgpt** to merge files with data from different days.

  - Y: Allow data from different dates to be merged together

    Mrgpt will use the environment variables `G_STDATE`, `G_STTIME`, `G_RUNLEN`, and `G_TSTEP` to set the start date, start time, length, and time step of the output file. All input files must have the same starting time as `G_STTIME`, the same time step as `G_TSTEP`, and must contain more than `G_RUNLEN` steps of data. For each input file, if it contains data to cover the actual episode requested by `G_STDATE`, `G_STTIME`, and `G_RUNLEN`, then the data for the actual episode will be used; otherwise, data starting at the first time step of the file will be used.

  - N: Require all merged data to be from the same date

    Mrgpt will determine the output file time period based on the latest starting input file and the earliest ending input file.

6.11.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Mrgpt**
OUTFILE_INLN	I/O API NetCDF	Merged inline-mode hourly speciated emission file
OUTFILE_STACK	ASCII	Merged inline-mode ASCII point elevated file
REPMRGGRID	ASCII	Report file generated when `MRG_DIFF_DAYS` is set to Y; the report lists the date and time for each output time step along with the actual date and time of the data read from each file
REPMERG_ADJ	ASCII	Report file generated when `ADJ_FACS` is set; the report lists the date, the logical file name, the species name, the factor applied, a daily total of species before and after applying adjustment factor, and the ratio (after/ratio). The report only lists when the ratio is not equal to 1.0

6.12. Normbeis3

6.12.1. Description

Processing categories: biogenic

Normbeis3 reads gridded land use data and emissions factors and produces gridded normalized biogenic emissions. **Normbeis3** is the first program in the SMOKE-BEIS3 model. The gridded land use includes various number of land use types. Emissions factors are provided for each land use type. The output gridded domain is the same as the input domain for the land use data. Emission fluxes are normalized to 30 °C, and isoprene and methyl-butenol fluxes are also normalized to a photosynthetic active radiation (PAR) of 1000 μmol/m^{2}s. **Normbeis3** also computes normalized emissions for winter months, using a winter factor read in from the emissions factor table.

For BEIS version 3.14 and 3.60, **Normbeis3** outputs winter and summer normalized emissions for the 35 species/compounds (including 14 monoterpenes and 1 sesquiterpene) listed below, as well as winter and summer leaf area indices (LAI). The output units for NO are grams of nitrogen per hour; the output units for LAI are meters squared per meters squared; the units for all other species/compounds are grams of carbon per hour.

Variable Name	Description
ISOP	Isoprene
MBO	2-methyl-3-buten-2-ol
METH	Methanol
ETHE	Ethene
PROPE	Propene
ETHO	Ethanol
ACET	Acetone
HEXA	Hexanal
HEXE	Hexenol
HEXY	Hexenylacetate
FORM	Formaldehyde
ACTAL	Acetaldehyde
BUTE	Butene
ETHA	Ethane
FORAC	Formic acid
ACTAC	Acetic acid
BUTO	Butenone (aka methyl vinyl ketone)
CO	Carbon monoxide
ORVOC	Other reactive VOCs
NO	Nitric oxide
LAI	Leaf area indices
SESQT	Sesquiterpene
MONO	*Monoterpenes*
APIN	Alpha-pinene
BPIN	Beta-pinene
D3CAR	Delta-3-carene
DLIM	D-limonene
CAMPH	Camphene

Variable Name	Description
MYRC	Myrcene
ATERP	Alpha-terpinene
BPHE	Beta-phellandrene
SABI	Sabinene
PCYM	P-cymene
OCIM	Ocimene
ATHU	Alpha-thujene
TRPO	Terpinolene
GTERP	Gamma-terpinene

6.12.2. Processing Order

Normbeis3 is the first program for processing biogenic emissions using the SMOKE-BEIS3 prototype. **Normbeis3** must be executed before **Tmpbeis3**.

6.12.3. Files and Environment Variables

Figure 6.12. Normbeis3 input and output files for BEIS v3.14
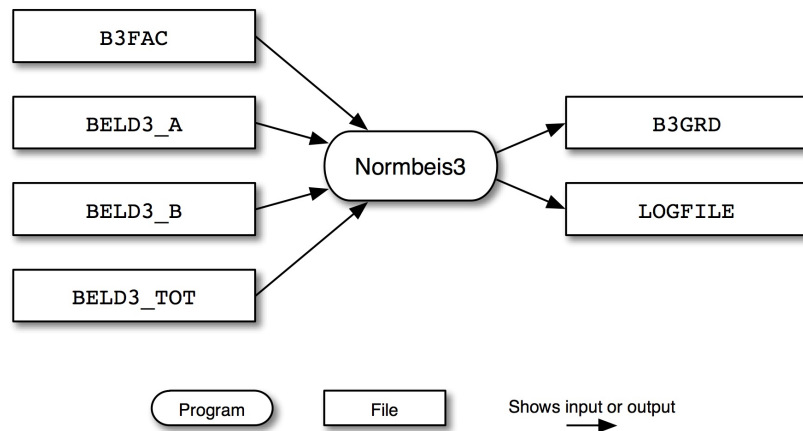
Figure 6.13, "**Normbeis3** input and output files for BEIS v3.60" [259] shows the input and output files for the **Normbeis3** program for BEIS v3.14. The input files are the emission factors file (B3FAC), the gridded land use file for the first 120 land use types (BELD3_A), the gridded land use types for the next 110 types (BELD3_B), and the land use totals data file (BELD3_TOT).

Figure 6.13. Normbeis3 input and output files for BEIS v3.60
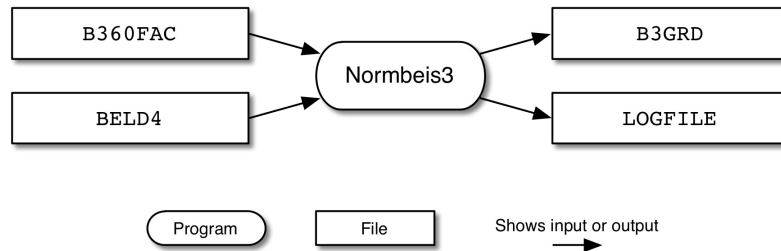
Figure 6.12, "**Normbeis3** input and output files for BEIS v3.14" [258] shows the input and output files for the **Normbeis3** program for BEIS v3.60. The input files are the emission factors file (`B360FAC`), the gridded land use file for the first 286 land use types (`BELD4`).

Normbeis3 outputs a gridded, normalized biogenic emissions file (`B3GRD`) and a log file (`LOGFILE`).

6.12.3.1. Input Files

Gridded land use data can be created using GIS software, such as ArcInfo. Another tool that can be used to generate gridded land use data is the SMOKE Tool in Models-3 (see Models-3 documentation). The typical database used to generate the `BELD3_A`, `BELD3_B`, and `BELD3_TOT` files is the BELD3. Future versions of SMOKE may include a method for creating land use files for use with **Normbeis3**.

File Name	Format	Description
`B3FAC`	ASCII	Emission factors file for BEIS v3.14
`BELD3_A`	I/O API NetCDF	Gridded land use (first 120 land use types) for BEIS v3.14
`BELD3_B`	I/O API NetCDF	Gridded land use (last 110 land use types) for BEIS v3.14
`BELD3_TOT`	I/O API NetCDF	Land use totals data for BEIS v3.14
`B360FAC`	ASCII	Emission factors file for BEIS v3.60
`BELD4`	ASCII	Gridded 286 land use for BEIS v3.60

6.12.3.2. Input Environment Variables

None

6.12.3.3. Output Files

File Name	Format	Description
`B3GRD`	I/O API NetCDF	Gridded, normalized biogenic emissions (in grams of carbon or nitrogen per hour, depending on the species) and LAI: contains both summer and winter normalized emissions
`LOGFILE`	ASCII	Log generated from executing **Normbeis3**

6.13. Smkinven

6.13.1. Description

Processing categories: area, mobile, point

Smkinven reads in the raw input data, sorts the records, and creates the SMOKE inventory files that are required by most of the SMOKE programs. The program is run separately for area-, mobile-, and point-source processing. **Smkinven** reads annual, average-day, day-specific, and hour-specific inventory files. The annual and average-day inventory data are combined into a single inventory. The output also indicates whether the data started out as annual or average-day data; such "temporal origin" information is needed by the **Temporal** program to determine whether to apply monthly adjustments. **Smkinven** can also import gridded one-layer annual or average-day emissions in I/O API NetCDF or CF-compliant NetCDF format as an area sourc.

Smkinven writes several SMOKE inventory files for each source category: an ASCII file containing all character-string fields from the inventory, individual I/O API NetCDF files containing emissions data for each output pollutant, and an ASCII map file listing the names and locations of each file. The character string fields cannot be stored in the I/O API NetCDF file because of limitations of the I/O API library. When day-specific and hour-specific data are available (this applies only for point sources), **Smkinven** can also read these data, and produce separate I/O API NetCDF files for each data type (day-specific or hour-specific).

Smkinven supports the following input file formats and source categories:

- SMOKE one-record-per-line (ORL) format for nonpoint, nonroad mobile, on-road mobile, and point sources

- SMOKE one-record-per-line (ORL) format for fire point sources (modified to support wildfires and prescribed fires characteristics and pollutants)

- Special mobile link format

- FF10 (Flat File 10) format for many different source categories including day-specific, hour-specific, monthly and annual for nonpoint, nonroad mobile, on-road mobile, and point sources (comma or semicolon delimited)

- ASCII Continuous Emissions Monitoring (CEM) format (hour-specific point-source emissions)

- List format (for combining multiple files of the same or compatible formats) for area, mobile, and point

- LIST GRID format (for reading multiple gridded one-layer annual CF-compliant NetCDF inventory data) as area sources

- Gridded one-layer annual or average-day NetCDF I/O API or CF-compliant NetCDF emissions data (must be treated as an area source)

It is not possible to import only day- or hour-specific data from FF10 input files. If only such data are to be used, mock annual files must be created; see Section 8.2.6, "PTDAY: Point source day-specific emissions" [367] and Section 8.2.7, "PTHOUR: Point source hour-specific emissions" [369] for more information on the required formats.

The list format is simply a list of file names with paths with a special format identifier that allows all files in the list to be processed together. To use a list format, all input files must have the same or compatible formats.

The list grid format is simply a list of CF-compliant NetCDF file names with paths with a special format identifier that allows all files in the list to be processed together. To use a list grid format, all input files must have the same or compatible formats with inventory data from the same year.

When using CEM data, you are encouraged to preprocess your data to prepare them down to the dates and/or sources in the domain. The sources in the grid will be selected based on matching to the inventory (if the inventory is for only states within the modeling grid) and/or by the **Grdmat** program, which will remove sources outside the grid from the model-ready results.

Smkinven also can process the hourly CEM data in a more sophisticated way. Hourly heat input from the CEM data are used to allocate annual emissons to hourly emission data. More information are available at (Section 5.3.4, "**CEMScan**" [190]) and (Section 2.9, "Inventory import" [53])

If new sources are added to the input files or sources are removed, **Smkinven** must be rerun for that type of source. However, **Smkinven** does not need to be rerun when processing additional control scenarios, growth projections, speciation profiles, or modeling grids.

By default, **Smkinven** will fill in missing average-day data with annual data when possible. To fill in missing annual data with average-day data, use the FILL_ANNUAL setting.

6.13.2. Processing Order

Smkinven must be run for a given processing type (i.e., area, mobile, point) before any other SMOKE programs for that processing type can be run.

When processing CEM data, the utility program **CEMScan** must be run before **Smkinven**.

6.13.3. Files and Environment Variables

Figure 6.14. Smkinven input and output files for area sources
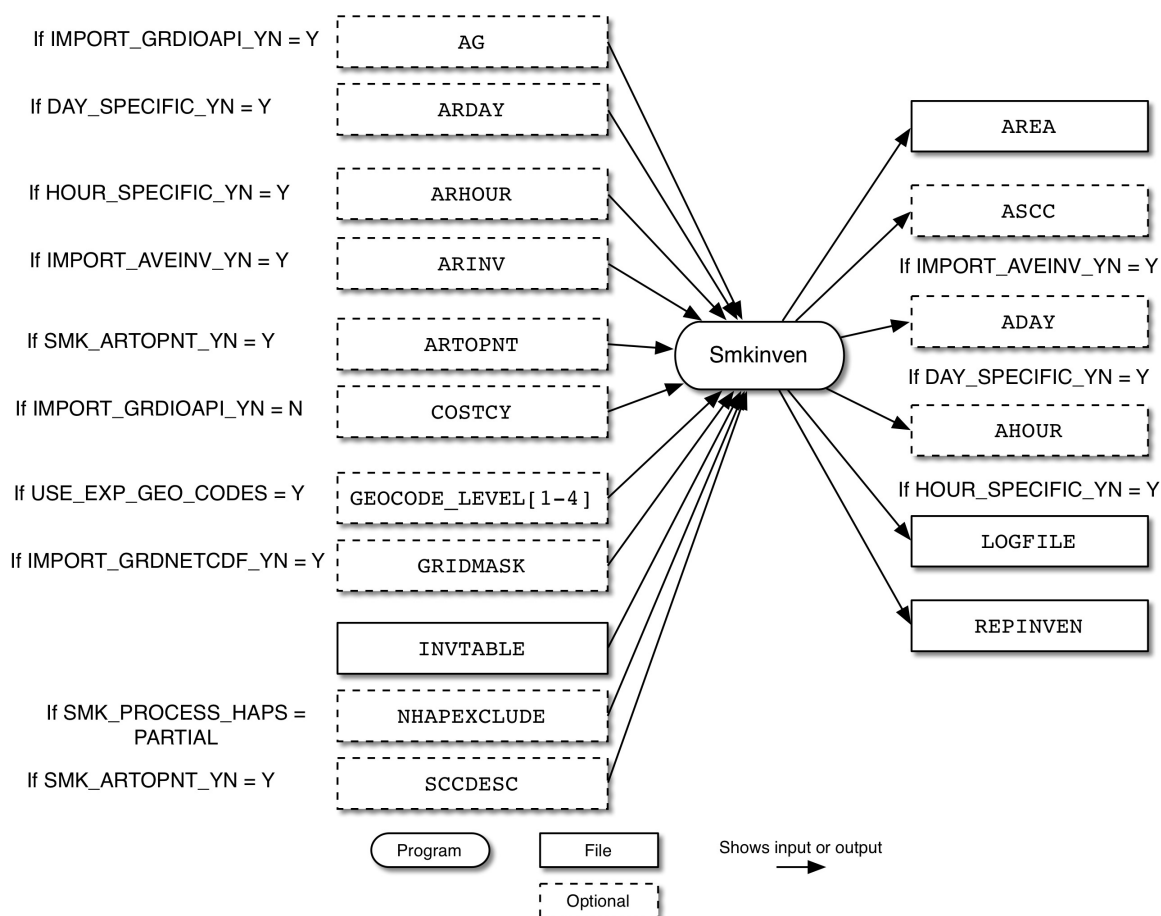
Figure 6.14, "**Smkinven** input and output files for area sources" [262] shows the input and output files for **Smkinven** for area sources. If importing gridded data, **Smkinven** reads the `AG` file; for the annual inventory, **Smkinven** reads the `ARINV` file. The `ARTOPNT` file is used to assign point-source locations to area sources, and either the `COSTCY` or `GEOCODE_LEVEL4` file is used to assign time zones to the annual inventory data. **Smkinven** also uses the inventory data table (`INVTABLE`), the optional NONHAP exclusions file (`NHAPEXCLUDE`), and the SCC description file (`SCCDESC`) if processing area-to-point sources.

Smkinven outputs an annual and average-day inventory emissions file (`AREA`), a file listing the inventory SCCs (`ASCC`), a log file (`LOGFILE`), and a report containing various information about the processed inventory (`REPINVEN`).

Figure 6.15. Smkinven input and output files for mobile sources
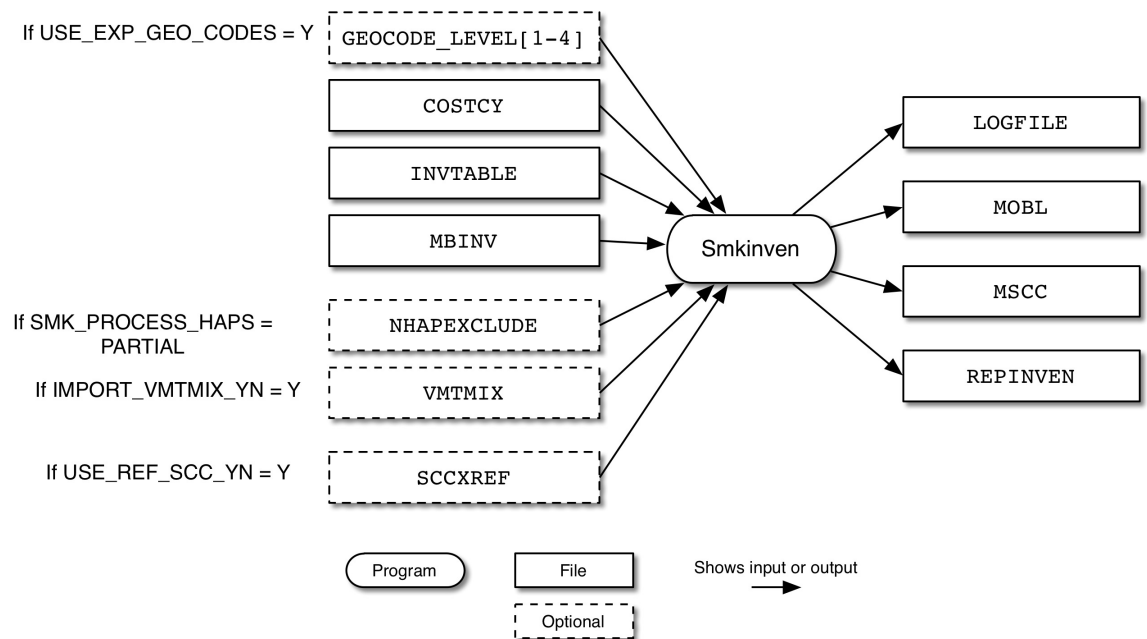
Figure 6.15, "**Smkinven** input and output files for mobile sources" [263] shows the input and output files for **Smkinven** for mobile sources. The input files include the country, state and county codes file (COSTCY), the inventory data table (INVTABLE), the raw inventory file (MBINV), and the SCC reference input file (SCCXREF) created by Gen_8digit_scc.pl.

Smkinven outputs a log file (LOGFILE), an annual and average-day inventory emissions file (MOBL), a file listing the inventory SCCs (MSCC), and a report containing various information about the processed inventory (REPINVEN).

Figure 6.16. Smkinven input and output files for point sources
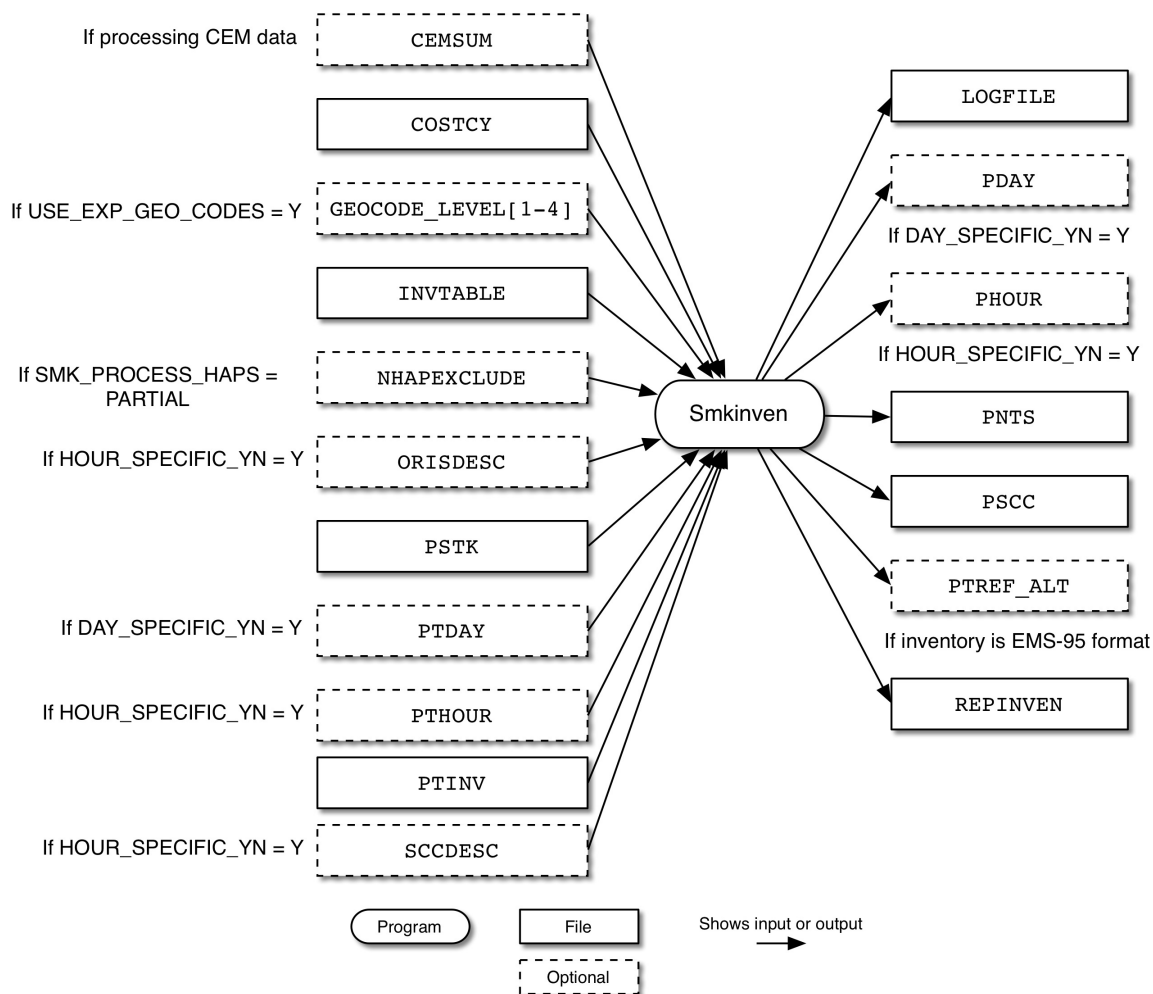
Figure 6.16, "**Smkinven** input and output files for point sources" [264] shows the input and output files for **Smkinven** for point sources. The input files include the optional CEM summary file (`CEMSUM`) created by **CEMScan**, the country, state and county codes file (`COSTCY`), the inventory data table (`INVTABLE`), and the optional NONHAP exclusions file (`NHAPEXCLUDE`). If importing hour-specific data, the ORIS facility description file `ORISDESC` is used. The `PSTK` file provides replacement stack parameters. Day-specific data is read from the `PTDAY` file, while hour-specific data comes from the `PTHOUR` file; the `PTINV` file provides the raw inventory data. For hour-specific data, the SCC description file (`SCCDESC`) is also needed.

Smkinven outputs a log file (`LOGFILE`), and optional day-specific inventory (`PDAY`), and an optional hour-spceific inventory (`PHOUR`). The annual and average-day inventory emissions are stored in the `PNTS` file and the `PSCC` file lists the inventory SCCs. Finally, the `REPINVEN` report contains various information about the processed inventory.

6.13.3.1. Input Files

File Name	Format	Source Type	Description
`AG` (optional)	I/O API NetCDF	A	Gridded annual or average-day data; this file can be created by **Smkmerge** and must be time-independent
`[AR|MB|PT]INV`	ASCII	A, M, P	Annual and average-day emissions and source characteristics raw inventory data
`ARTOPNT` (optional)	ASCII	A	Area-to-point cross-reference file for assigning X and Y locations to area sources
`CEMSUM` (optional)	ASCII	P	CEM summary data produced by **CEMScan**
`COSTCY`	ASCII	A, M, P	Country, state, and county codes file, with time zones
`GEOCODE_LEVEL[1-4]`	ASCII	A, B, M, P	Geographical level codes and descriptions for gridded inventories, only when USE_EXP_GEO_CODES is set to Y, (COSTCY is ignored when `GEOCODE_LEVEL[1-4]` files are used)
`GRIDMASK`	I/O API NetCDF	A, B, M, P	Defines an integer geocode to specify the predominant country and state code covered by each grid cell (`GEOCODE_LEVEL`) and an integer `TZONES` to specify the predominant Time Zone for each grid cell.
`INVTABLE`	ASCII	A, M, P	Inventory data table listing pollutant and activity names
`NHAPEXCLUDE` (optional)	ASCII	A, M, P	List of counties and SCCs to exclude from toxics integration
`ORISDESC` (optional)	ASCII	P	ORIS facility descriptions, needed for getting ORIS facility names for reporting purposes when CEM hour-specific data are read
`PSTK`	ASCII	P	Replacement stack parameters file
`PTDAY` (optional)	ASCII	P	Day-specific raw inventory data
`PTHOUR` (optional)	ASCII	P	Hour-specific raw inventory data
`SCCDESC` (optional)	ASCII	P	SCC descriptions, needed for reading CEM hour-specific data only, to select electric generator facilities in the inventory
`SCCXREF` (optional)	ASCII	M	SCC cross reference input file created by `Gen_8digit_scc.pl`. It maps aggregated mobile activity SCC8 to full SCC10.

6.13.3.2. Input Environment Variables

- `USE_EXP_GEO_CODES`: [default: N]

  If set to Y then use GEOCODE_LEVEL[1-4] files instead of the COSTCY file to define expanded geographical detail such as country-level(GEOCODE_LEVEL1), country/state-level (GEO-

CODE_LEVEL2), country/state/county-level (GEOCODE_LEVEL3) or country/state/county/tribal-level (GEOCODE_LEVEL4).

- ALLOW_NEGATIVE: [default: N]

  Indicates if **Smkinven** should permit negative emissions values to be input from inventory files and output to SMOKE intermediate files.

  - Y: Accept negative emissions values and will allow compute variables (from the SMKINVEN_FOR-MULA variable) to be negative. **Smkinven** gives warnings for each negative value in the inventory input and each computed value, until triggering the SMK_MAXWARNING setting's values.

  - N: Do not permit negative emissions values. Accept negative emissions values in the input inventory files, and will give errors for these. **Smkinven** will reset all negative computed values (from the SMKINVEN_FORMULA variable) to zero. **Smkinven** gives warnings for each negative computed value and indicates that the value was reset to zero.

- CHECK_STACKS_YN: [default: Y]

  Indicates if stack parameter values should be checked for missing or out-of-range values; only checked if SMK_SOURCE is set to P.

  - Y: Check input stack parameters for missing or out-of-range values

  - N: Do not check input stack parameter values

- DAY_SPECIFIC_YN: [default: N]

  Defines whether to import day-specific inventory data.

  - Y: Import day-specific data

  - N: Do not import day-specific data

- EXCLUDE_REF_SCC_YN: [default: N]

  Defines whether to exclude a list of SCCs that are not listed in Section 8.10.11, "SCCXREF: Reference SCC input file" [457]. This option is only applicable when USE_REF_SCC_YN is set to Y.

  - Y: Import day-specific data

  - N: Do not import day-specific data

- FF10_AVEDAY_ANNINV_YN: [default N]

  Uses area daily or hourly FF10-formatted invenotry as annual data. This will allow users to process ARDAY or ARHOUR invenotry data independently without artifical annual inventory data.

  - Y: Use daily or hourly FF10-formatted inventory as annual data.

  - N: Do not use daily or hourly FF10-formatted inventory as annual data. Just it as daily or hourly FF10-formatted inventory data.

- FILL_ANNUAL: [default N]

  Allows annual data to be filled in with average-day data.

- Y: Use "Y" when annual emission numbers are blank for some records of the inventory but not all records. The emissions values will be filled in using the average-day emissions field, by multiplying the average-day number by the number of days in the inventory year (365 or 366). This option is useful when some sources have only annual data and some sources have only average-day data. Although the annual number will be filled in using the average-day one, the average-day emissions will be output from SMOKE as a whole.

  - N: Do not fill in annual data based on average-day emissions values.

- `FLOW_RATE_FACTOR`: [default: 0.0 ft^3/MMBTU]

  Used to calculate hourly flow rates from hourly heat input when reading CEM data. If `FLOW_RATE_FACTOR` is set to zero or unset, then **Smkinven** will not calculate hourly flow rates. Detail information is described in Section 2.9.19, "Processing hour-specific CEM data" [62]

  hourly flow rate (m^3/s) = [`FLOW_RATE_FACTOR` (ft^3/MMBTU) * hourly heat input (MMBTU/hr) * 0.02831 m^3/ft^3] / 3600 s/hr

- `FULLSCC_ONLY`: See description in Section 6.2.3.2, "Input Environment Variables" [221]. In this context, this option affects cross-referencing for the `ARTOPNT`, and `NHAPEXCLUDE` files by preventing left-7 SCC assignments. It also prevents left-2 and left-4 SCC assignments for NHAPEXCLUDE.

- `HOUR_SPECIFIC_YN`: [default: N]

  Defines whether to import hour-specific inventory data.

  - Y: Import hour-specific data

  - N: Do not import hour-specific data

- `HOURLY_TO_DAILY`: [default: N]

  Defines whether to treat the hour-specific inputs as day-specific data.

  - Y: Use daily-total column only from hour-specific input file, and process as day-specific data. *This setting does not work for input of hour-specific data using CEM format.*

  - N: Treat hourly file as hourly data

- `HOURLY_TO_PROFILE`: [default: N]

  Defines whether to treat the hour-specific inputs as hour-specific temporal profiles.

  - Y: Treat hourly data as temporal profile, and divide each hourly value by value in the daily total column of the file. The values are not renormalized. The resulting hour-specific output file will be recognized by the **Temporal** program as containing hourly profiles. *This setting does not work for input of hour-specific data using CEM format.*

  - N: Treat hourly file as hourly data

- `IMPORT_AVEINV_YN`: [default: Y]

  Defines whether to import the annual and average-day inventory data. This is set to N to import day-specific or hour-specific data after **Smkinven** has previously been run for the annual and average-day data.

  - Y: Import annual and average-day inventory data from `PTINV` file

- N: Use SMOKE inventory files from a previous run of **Smkinven** to get the SMOKE source characteristics to match with the day-specific and hour-specific sources

- `IMPORT_GRDIOAPI_YN`: [default: N]

  Defines whether the program will import a gridded I/O API NetCDF inventory file. This setting can be used only with area-source processing, and it overrides all other program settings for importing data (e.g., day- and hour-specific data import will be shut off).

  - Y: Import gridded I/O API data

  - N: Do not import gridded I/O API data

- `INVNAME1`: [default: none]

  Sets root for naming output map inventory file

- `INVNAME2`: [default: none]

  Sets root for naming output ASCII inventory file

- `NONHAP_TYPE`: [default: VOC]

  Define the name of organic compound inventory pollutant VOC or TOG for a proper integration between VOC or TOG, and hazardous air pollutants (HAPs) to compute NONHAPVOC or NONHAP-TOG. See Section 2.9.5, "Combine toxics and criteria inventories" [55] for a more detail description of criteria/HAP VOC integration. This option will be used only when `SMK_PROCESS_HAPS` is set.

- `RAW_DUP_CHECK`: [default: N]

  Defines whether the program checks for duplicate records in the input file.

  - Y: Check for duplicates, and produce a fatal error if found

  - N: Do not check for duplicates, and add emissions from duplicate records

- `SMK_ARTOPNT_YN`: [default: N]

  - Y: Assign X and Y coordinates to specified sources based on the `ARTOPNT` cross-reference file

  - N: Do not perform area-to-point processing

- `SMK_BASEYR_OVERRIDE`: [default: 0]

  Used to override the base year determined by the inventory.

- `SMK_DEFAULT_TZONE`: [default: 5]

  Defines the default time zone to use for sources that do not match entries in the `COSTCY` or the `GEO-CODE_LEVEL4` file.

- `SMK_MAXERROR`: [default: 100]

  Defines the maximum number of ERROR statements to be displayed by SMOKE.

- `SMK_MAXWARNING`: [default: 100]

  Defines the maximum number of WARNING statements to be displayed by SMOKE.

- `SMK_PROCESS_HAPS`: [default: blank]

  Defines how to treat sources for combining criteria VOC and hazardous air pollutants (HAP) together. See Section 2.9.5, "Combine toxics and criteria inventories" [55] for more detail.

  - ALL: Treat all sources as integrated if a toxics inventory has been provided. Does not require an `NHAPEXCLUDE` file

  - NONE: Treat all sources as non-integrate if a toxics inventory has been provided. Does not require an `NHAPEXCLUDE` file

  - PARTIAL: Read the `NHAPEXCLUDE` cross-reference file and either include or exclude the specified sources from the NONHAP[VOC|TOG] calculation depending on the header of the `NHAPEXCLUDE` file. See Section 8.10.7, "NHAPEXCLUDE: NonHAP VOC calculation inclusions/exclusions file" [453].

- `SMK_SOURCE`: See description in Section 6.2.3.2, "Input Environment Variables" [221]

- `SMK_TMPDIR`: [default: `.`]

  Sets the path for writing temporary files needed when importing the inventory.

- `SMKINVEN_FORMULA`: [default: blank]

  Allows the user to specify a formula for computing one or more pollutant values from the values of other pollutants. To use, users should set the value of `SMKINVEN_FORMULA` to a double-quote-enclosed string of 512 characters or less, using a comma-delimited list of formulas. No spaces are necessary in the formula list. Only formulas of a specific structure are accepted. Specifically, one of the following:

  - OUTPUT = INPUT1 + INPUT2

  - OUTPUT = INPUT1 - INPUT2

  For example, if the formula is set to "PMC=PM10-PM2_5,EXH__PMC=EXH__PM10-EXH__PM2_5", then **Smkinven** will compute two new variables: (1) PMC, as the difference between PM_{10} and $PM_{2.5}$ and (2) EXH__PMC as the difference between exhaust PM_{10} and exhaust $PM_{2.5}$. The pollutant units are taken from the first variable in each formula. The variables used as input in the formula will still be output to the SMOKE inventory file. If the formula is not set, no additional variables will be written to the SMOKE inventory file. An output from one formula may not be used as an input to another formula. In the OUTPUT pollutant name is the same as an INPUT pollutant name from the inventory file, Smkinven will give an error indicating that the second pollutant could not be written (e.g. if computing PMC when PMC is already in the raw inventory data). If INPUT2 is ommitted, an error will also occur. Formulas using multiplication and division are not yet implemented.

- `SMKINVEN_MONTH`: [default: 0]

  Defines the number of month to process. The integer month number will allow users to pick monthly total inventory data in area FF10-formatted invenotry data listed at Table 8.15, "FF10 Format for `AR-INV`" [357].

- `USE_REF_SCC_YN`: [default: N]

  Defines whether use reference SCC (i.e., SCC8) to full SCC (i.e., SCC10) mapping input file to convert reference SCC-level inventory to full SCC inventory. See detail from Section 8.10.11, "`SCCXREF`: Reference SCC input file" [457]

- Y: Use reference SCC mapping input file to convert reference SCCs to full SCCs.

- N: Do not use SCCXREF input file.

- `VELOC_RECALC`: [default: N]

  Defines whether stack velocity is recalculated in order to make the velocity consistent with the reported flow and diameter.

  - Y: Recalculate stack velocity

  - N: Output the input stack velocity

- `WEST_HSPHERE`: [default: Y]

  Defines how spatial coordinates are interpreted.

  - Y: Convert longitudinal coordinates for the Western Hemisphere (positive values are converted to negative values)

  - N: Do not convert longitude values (positive values are not changed)

- `WKDAY_NORMALIZE`: [default: N]

  Clarifies the nature of the "average-day" data read in by **Smkinven** and will impact the **Temporal** program processing of average-day data for all formats.

  - Y: Average-day emissions input to **Smkinven** are based on weekday emissions only. In **Temporal**, these emissions will be adjusted with weekly profiles normalized with weekdays only (Monday through Friday)

  - N: Average-day emissions input to **Smkinven** are based on *all days of the week*. In **Temporal**, these emissions will be adjusted with weekly profiles normalized with all days of the week.

- `WRITE_ANN_ZERO`: [default: N]

  Indicates if annual emission values equal to zero should be written to the intermediate inventory file

  - Y: Write zero emission values; this option makes the intermediate inventory file larger but is sometimes necessary when only daily or hourly data is provided and the entries in the annual inventory are dummy values

  - N: Do not write zero values

6.13.3.3. Output Files

File Name	Format	Description
`[AREA|MOBL|PNTS]`	SMOKE map	Annual and average-day inventory emissions and source characteristics
`[ASCC|MSCC|PSCC]`	ASCII	List of SCCs appearing in the raw inventory file
`LOGFILE`	ASCII	Log generated from executing **Smkinven**
`ADAY	PDAY`	I/O API NetCDF	Day-specific inventory emissions assigned to sources in annual/average-day files (optional)

File Name	Format	Description
`AHOUR	PHOUR`	I/O API NetCDF	Hour-specific inventory emissions assigned to sources in annual/average day files (optional)
`REPINVEN` (optional)	ASCII	Inventory report file; contents depend on the inventory being processed, but can include: CEM section: • ORIS IDs from the CEM matched to the inventory • ORIS IDs from the CEM not matched to the inventory • Power-generating inventory sources not matched to the CEM Toxics section: • Total emissions and number of inventory records by CAS number • Total emissions before and after aggregation by CAS number Area-to-point section: • SCCs in the `ARTOPNT` file but not in the inventory • Total emissions before and after area-to-point processing by pollutant, state, and SCC

6.14. Smkmerge

6.14.1. Description

Processing categories: area, biogenic, mobile, point

The **Smkmerge** program combines the matrices produced by the other SMOKE programs to produce the emissions files for input to the AQM. The **Smkmerge** program may be run on any combination of source types and may incorporate temporal, speciation, projection, and spatial processing.

If **Smkmerge** is run to produce the input files to the AQM, all the other SMOKE programs must already have been run. However, you can also run **Smkmerge** to produce specialized reports. In that instance, only the required subset of SMOKE programs must have been run prior to **Smkmerge**.

Smkmerge can be used to create 2-D or 3-D gridded, hourly, speciated files. It also outputs the hourly emissions file for PinG sources needed by CMAQ and MAQSIP. Finally, it outputs an ASCII elevated point sources file, suitable for postprocessing with the Ptsrce program of EPS2.0/EPS2.5. If PinG sources have been selected and an ASCII elevated sources file is output, then SMOKE will set the stack diameters to negative values for the PinG sources.

6.14.1.1. By-day processing

Smkmerge can use a by-day processing approach, to read in one-day hourly files for each day of the week. When the `MRG_BYDAY` environment variable is set to A, M, P, or any combination of those letters, **Smkmerge** reads a different logical file name for each day of the week for the hourly emission input files for the indicated sources.

For example, if the `MRG_SOURCE` environment variable is set to AMBP, **Smkmerge** will expect the following hourly input files: `ATMP`, `BGTS_[S|L]`, `MTMP`, and `PTMP`, for area, biogenic, mobile, and point source types respectively. If, in addition, `MRG_BYDAY` is set to "A", then the hourly input files expected by **Smkmerge** for area sources would be the `ATMP_{day}` files. The files for the other source types would remain the same.

The `MRG_BYDAY` option is useful for implementing a Monday-weekday-Saturday-Sunday (MWSS) approach to save processing time for long episodes. This approach is applicable when weekly temporal profiles used by an inventory do not have variation between the five weekdays. In the MWSS approach, Monday is processed separately, because time zone differences across a large domain cause some Sunday emissions for some time zones to be in the Monday file. Additional environment variables other than the defaults in the Assigns file need to be set when this option is applied. **Smkmerge** will prompt for a different file for each day of the week, but the logical file names for Tuesday through Friday can be defined as the same file when implementing the MWSS approach.

6.14.2. Processing Order

All other SMOKE core programs except the optional program **Mrggrid** prepare input for the **Smkmerge** program. **Smkmerge** is the last nonoptional SMOKE program to be run.

Individual output files from **Smkmerge** can be merged using **Mrggrid** if desired.

6.14.3. Files and Environment Variables

Figure 6.17. Smkmerge input and output files for area sources
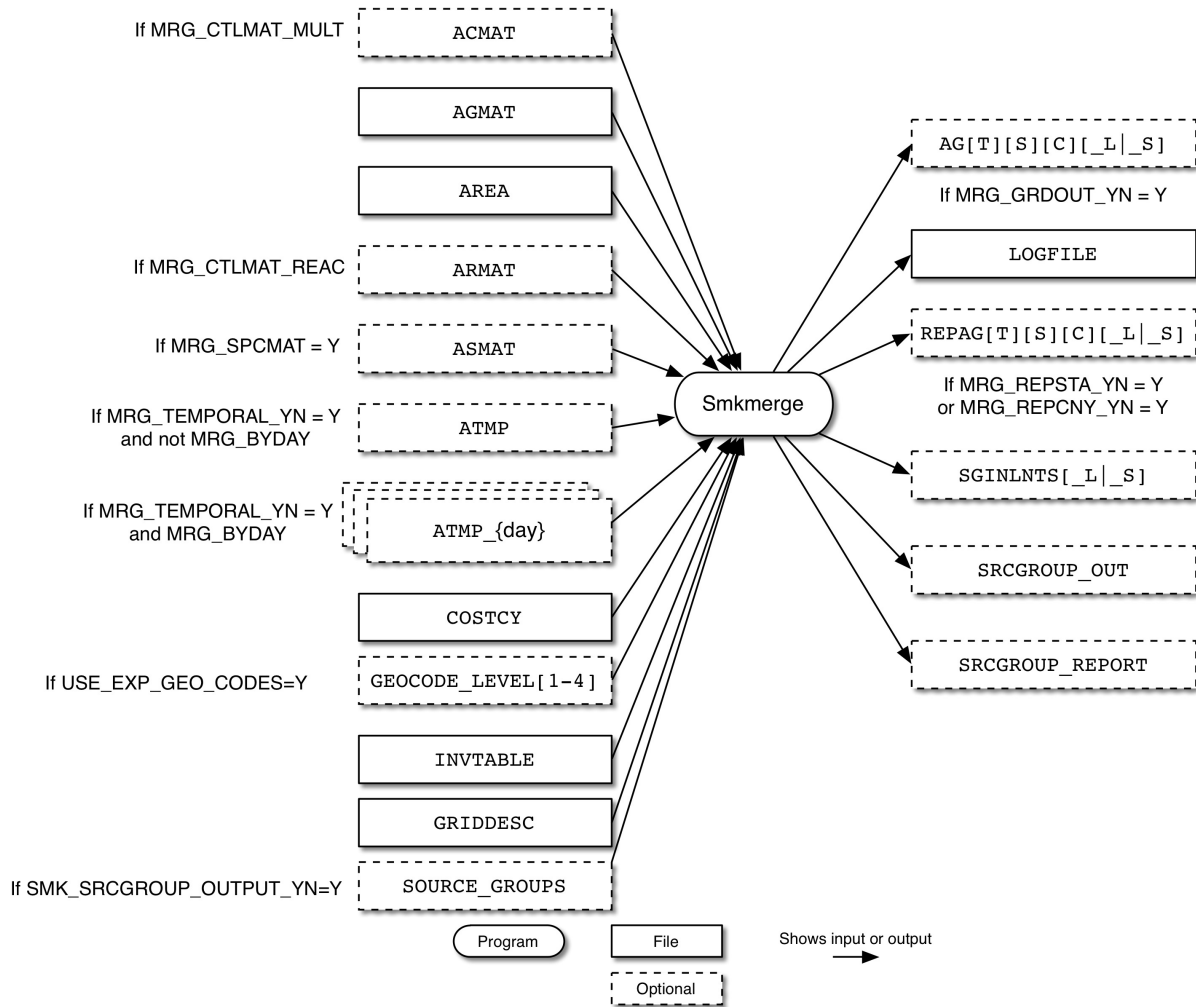
Figure 6.17, "**Smkmerge** input and output files for area sources" [273] shows the input and output files for the **Smkmerge** program for area sources. The inputs include the optional multiplicative control matrix from **Cntlmat** (ACMAT), the gridding matrix from **Grdmat** (AGMAT), the inventory file output from **Smkinven** (AREA), the optional reactivity matrix from **Cntlmat** (ARMAT), and the optional speciation matrix from **Spcmat** (ASMAT). **Smkmerge** can also use hourly emissions from **Temporal**, either a single file (PTMP) or hourly emissions for each day of the week (ATMP_{day}). The last three input files are the country, state, and county codes file (COSTCY), the inventory data table (INVTABLE), and the grid description file (GRIDDESC).

Smkmerge outputs a data file that is determined by the inputs and can be a model-ready emissions file (AG[T][S][C]_L or AG[T][S][C]_S), a log file (LOGFILE), and an optional report file (RE-PAG[T][S][C]_L or REPAG[T][S][C]_S).

When using source apportionment (see Section 4.4.23, "Output source apportionment data" [175]), **Smkmerge** will also read the source grouping file (SOURCE_GROUPS) and output the source group information SR-

CGROUPS_OUT, the emissions for each source group SGINLNTS_L or SGINLNTS_S, and a source group report file SRCGRP_REPORT.

Figure 6.18. Smkmerge input and output files for biogenic sources

If SMK_SRCGROUP_OUTPUT_YN=Y

Figure 6.18, "**Smkmerge** input and output files for biogenic sources" [274] shows the input and output files for the **Smkmerge** program for biogenic sources. The inputs are the optional surrogates file (BGPRO), the model-ready emissions file (BGTS), the country, state, and county file (COSTCY), the grid description file (GRIDDESC), and the inventory data table (INVTABLE).

Smkmerge outputs either a mole-based emissions file (BGTS_L_O) or a mass-based emissions file (BGTS_S_O). Note that _O is appended to the output file name to distinguish it from the input file. The program also writes a log file (LOGFILE) and an optional report file (REPBGTS_L or REPBGTS_S).

When using source apportionment (see Section 4.4.23, "Output source apportionment data" [175]), **Smkmerge** will also read the source grouping file (SOURCE_GROUPS) and output the source group information SR-CGROUPS_OUT, the emissions for each source group SGINLNTS_L or SGINLNTS_S, and a source group report file SRCGRP_REPORT.

Figure 6.19. Smkmerge input and output files for mobile sources
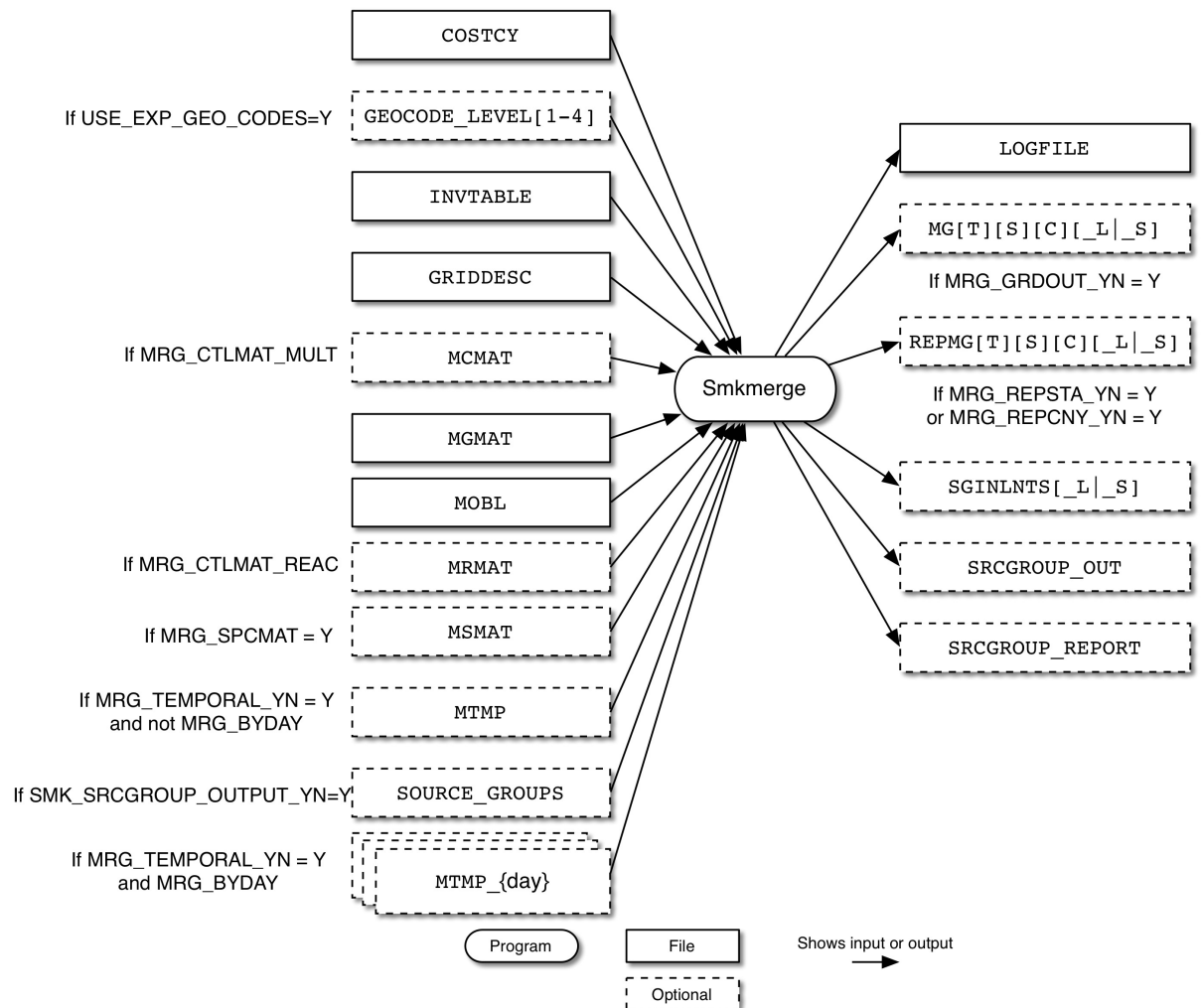
Figure 6.19, "**Smkmerge** input and output files for mobile sources" [275] shows the input and output files for the **Smkmerge** program for mobile sources. The inputs include the country, state and county codes file (COSTCY), the inventory data table (INVTABLE), and the grid description file (GRIDDESC). **Smkmerge** also reads the optional multiplicative control matrix from **Cntlmat** (MCMAT), the gridding matrix from **Grdmat** (MGMAT), the inventory file output from **Smkinven** (MOBL), the optional reactivity matrix from **Cntlmat** (MRMAT), and the optional speciation matrix from **Spcmat** (MSMAT_L or MSMAT_S). Hourly emissions can be input either as a single file (MTMP) or as hourly emissions for each day of the week (MTMP_{day}).

Smkmerge outputs a log file (LOGFILE), a data file that is determined by the inputs and can be a model-ready emissions file (MG[T][S][C]_L or MG[T][S][C]_S), and an optional report file (REPMG[T][S][C]_L or REPMG[T][S][C]_S).

When using source apportionment (see Section 4.4.23, "Output source apportionment data" [175]), **Smkmerge** will also read the source grouping file (SOURCE_GROUPS) and output the source group information SRCGROUPS_OUT, the emissions for each source group SGINLNTS_L or SGINLNTS_S, and a source group report file SRCGRP_REPORT.

Figure 6.20. Smkmerge input and output files for point sources
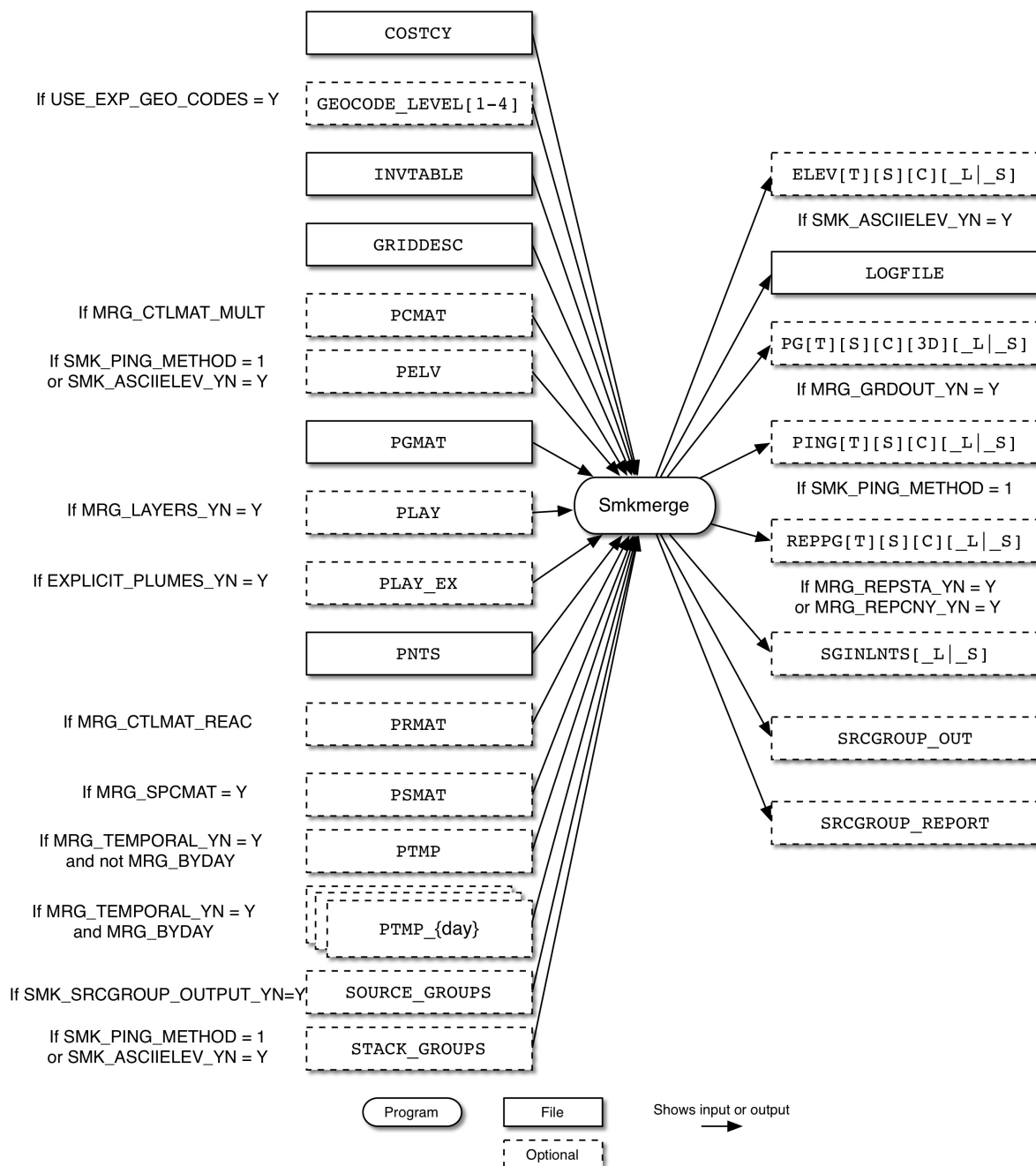
Figure 6.20, "**Smkmerge** input and output files for point sources" [276] shows the input and output files for the **Smkmerge** program for point sources. The inputs include the country, state, and county codes file (`COSTCY`), the inventory data table (`INVTABLE`), the grid description file (`GRIDDESC`), and the optional multiplicative control matrix from **Cntlmat** (`PCMAT`). If using PinG or elevated sources, the `PELV` file is used. **Smkmerge** reads the gridding matrix output from **Grdmat** (`PGMAT`). The `PLAY` file is used to merge layer fractions, while the `PLAY_EX` file is used for explicit plumes. Other **Smkmerge** input files include the inventory file output from **Smkinven** (`PNTS`), the optional reactivity matrix from **Cntlmat** (`PRMAT`), and the optional speciation matrix from **Spcmat** (`PSMAT`). The hourly emissions from **Temporal** are

provided either as the `PTMP` file or as hourly emissions for each day of the week (`PTMP_{day}`). Finally, **Smkmerge** can read the optional elevated stack groups file (`STACK_GROUPS`).

Smkmerge can output an elevated sources file (`ELEV[T]S[_S|_L]`) if requested. Like all other SMOKE programs, **Smkmerge** outputs a log file (`LOGFILE`) detailing the program's execution. The main data file is determined by the inputs and can be include a model-ready emissions file (`PG[T][S][C][3D]_L` or `PG[T][S][C][3D]_S`). Other output files can be a PinG file (`PING[T][S][C][_S][_L]` and a report file (`REPPG[T][S][C]_L` or `REPPG[T][S][C]_S`).

When using source apportionment (see Section 4.4.23, "Output source apportionment data" [175]), **Smkmerge** will also read the source grouping file (`SOURCE_GROUPS`) and the stack groups file (`STACK_GROUPS`), and output the source group information `SRCGROUPS_OUT`, the emissions for each source group `SGIN-LNTS_L` or `SGINLNTS_S`, and a source group report file `SRCGRP_REPORT`.

Figure 6.21. Smkmerge input and output files for all source categories
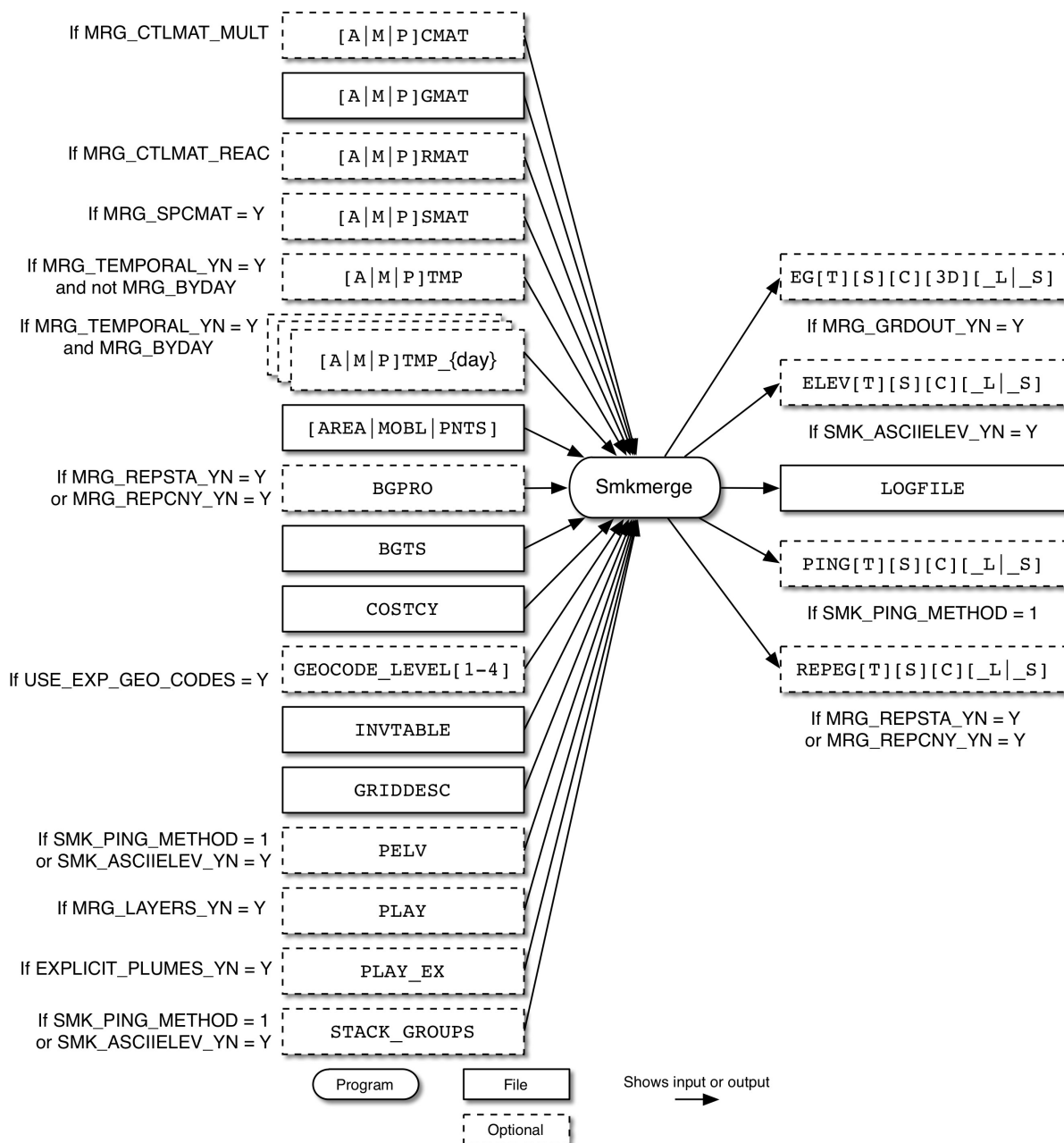
Figure 6.21, "**Smkmerge** input and output files for all source categories" [278] shows the input and output files for the **Smkmerge** program for all sources. The input and output files are a combination of those found in the previous four figures.

6.14.3.1. Input Files

The input files needed depend upon the input environment variables that are set for this program. For example, if only area sources are being merged, then only area-source-related inputs are required. Similarly, if **Smkmerge** is being executed to produce speciated, gridded, temporally allocated area and point sources,

then all area- and point-related inputs for area and point sources, speciation matrices, gridding matrices, and temporal matrices are required.

File Name	Format	Source Type	Description
`[A|M|P]CMAT` (optional)	I/O API NetCDF	A, M, P	Multiplicative control matrix, output by **Cntlmat** (used when `MRG_CTLMAT_MULT` is set)
`[A|M|P]GMAT`	I/O API NetCDF	A, M, P	Gridding matrix from **Grdmat**
`[A|M|P]RMAT` (optional)	I/O API NetCDF	A, M, P	Mass- or mole-based reactivity matrix, output by **Cntlmat** (used when `MRG_CTL-MAT_REAC` is set)
`[A|M|P]SMAT` (optional)	I/O API NetCDF	A, M, P	Mass- or mole-based speciation matrix from **Spcmat** (used when `MRG_SPCMAT_YN = Y`)
`[A|M|P]TMP` (optional)	I/O API NetCDF	A, M, P	Hourly emissions from **Temporal** (used when `MRG_TEMPORAL_YN = Y` and `MRG_BYDAY` is not set)
`[A|M|P]TMP_{day}` (optional)	I/O API NetCDF	A, M, P	Hourly emissions from **Temporal** by day, where {day} = MON, TUE, WED, THU, FRI, SAT, SUN (used when `MRG_TEMPOR-AL_YN = Y` and `MRG_BYDAY` is set)
`[AREA|MOBL|PNTS]`	SMOKE map	A, M, P	Intermediate area-, mobile-, or point-source inventory created by **Smkinven**
`BGPRO` (optional)	ASCII	B	Gridding surrogates for computing biogenic state and county totals from gridded biogenic inputs
`BGTS`	I/O API NetCDF	B	Biogenic model-ready emissions, to be input only when merging with other source categories, or when computing biogenic state and county totals, produced by **Tmpbio** or **Tmpbeis3**
`COSTCY`	ASCII	A, B, M, P	Country, state, and county file with time zones
`GEOCODE_LEVEL[1-4]`	ASCII	A, B, M, P	Geographical level codes and descriptions for gridded inventories, only when `USE_EXP_GEO_CODES` is set to Y, (`COSTCY` is ignored when `GEO-CODE_LEVEL[1-4]` files are used)
`INVTABLE`	ASCII	A, B, M, P	Inventory data table containing activity and pollutant names
`GRIDDESC`	ASCII	A, B, M, P	Description of the modeling grid
`PELV` (optional)	ASCII	P	Elevated point sources identification file, output by **Elevpoint**
`PLAY` (optional)	I/O API NetCDF	P	Hourly plume rise layer fractions, output by **Laypoint** (used when `MRG_LAYERS_YN = Y` and `EXPLICIT_PLUMES_YN = N`)

File Name	Format	Source Type	Description
PLAY_EX (optional)	I/O API NetCDF	P	Hourly plume rise layer fractions for explicit sources, output by **Laypoint** (used when MRG_LAYERS_YN = Y and EXPLI-CIT_PLUMES_YN = Y)
STACK_GROUPS (optional)	I/O API NetCDF	P	Elevated sources stack parameters file, output by **Elevpoint**
SOURCE_GROUPS (optional)	ASCII	A, B, M, P	Source apportionment groups

6.14.3.2. Input Environment Variables

NOTE: At least one of the MRG_GRDOUT_YN, MRG_REPSTA_YN, or MRG_REPCNY_YN environment variables must be set to Y.

- AREA_SURROGATE_NUM: [default: 60]

   Surrogate code of the area surrogate in the BGPRO file, to use for totals of biogenic emissions by state and county.

- EXPLICIT_PLUMES_YN: [default: N]

   Indicates whether only explicit plume rise sources should be merged.

   - Y: Merge only explicit plume rise sources; uses PLAY_EX file

   - N: Merge all plume rise sources; uses PLAY file

- USE_EXP_GEO_CODES: [default: N]

   If set to Y, then use GEOCODE_LEVEL[1-4] files instead of the COSTCY file to define expanded geographical detail such as country-level(GEOCODE_LEVEL1), country/state-level (GEO-CODE_LEVEL2), country/state/county-level (GEOCODE_LEVEL3) or country/state/county/tribal-level (GEOCODE_LEVEL4).

- MRG_BYDAY: [default: blank]

   String code that specifies which source categories (if any) are to use a by-day approach for the hourly input files (see "Description" section above for more information on this approach). Valid settings include A, M, P, or any combination. For example, set this environment variable to "AMP" to have **Smkmerge** read the area-, mobile-, and point-source hourly emissions files individually for each day.

- MRG_CTLMAT_MULT: [default: blank]

   String code that specifies which source categories (if any) are to be processed using a multiplicative control matrix. Valid settings include A, M, P, or any combination. For example, set this environment variable to "M" to have **Smkmerge** read the mobile-source multiplicative control matrix.

- MRG_CTLMAT_REAC: [default: blank]

   String code that specifies which source categories (if any) are to be processed using a reactivity control matrix. Valid settings include A, M, P, or any combination. For example, set this environment variable to "MP" to have **Smkmerge** read the mobile- and point-source reactivity control matrices.

- `MRG_GRDOUT_UNIT`: [default: input file units]

  String code that permits the user to change the units of the gridded output file. For example, "kg/hr" would convert the output emissions to kilograms per hour, before outputting the emissions data file. Not all conversions are known to SMOKE. Valid numerators are: "tons", "kg", "moles", and "g"; note that molar emissions cannot be output without the mole-based speciation matrices. Valid denominators are "yr", "dy", "hr", and "s". Note that if annual output is requested but not available in the input file, leap year adjustments are not made.

  Units required by CMAQ are moles/s and units needed for MAQSIP and most other air quality models are moles/hr.

- `MRG_GRDOUT_YN`: [default: N]

  Defines whether the program produces a gridded output file.

  - Y: Produce the file

  - N: Do not produce the file

- `MRG_LAYERS_YN`: [default: N]

  Defines whether the program uses the layer fractions file.

  - Y: Use the layer fractions file

  - N: Do not use the layer fractions file

- `MRG_MARKETPEN_YN`: [default: Y]

  Defines whether the program uses market penetration from reactivity matrices.

  - Y: Use market penetration

  - N: Do not use market penetration

- `MRG_METCHK_YN`: [default: Y]

  Defines whether the program checks consistency of headers in the meteorology file.

  - Y: Check contents of meteorology file headers

  - N: Do not check headers

- `MRG_REPCNY_YN`: [default: N]

  Defines whether the program produces a report of emission totals by county.

  - Y: Produce county-total emissions report

  - N: Do not produce the report

- `MRG_REPCTL_YN`: [default: N]

  Defines whether the program reports controlled emissions or not.

  - Y: Report controlled emissions

- N: Do not report controlled emissions

- `MRG_REPSTA_YN`: [default: N] Defines whether the program produces a report of emission totals by state.

    - Y: Produce state-total emissions report

    - N: Do not produce the report

- `MRG_SOURCE`: [default: blank (processing halts)]

  Series of character codes representing the type(s) of sources to be merged.

    - A: Area sources

    - B: Biogenic sources

    - M: Mobile sources

    - P: Point sources

    - AP: Area and point sources

    - AM: Area and mobile sources

    - etc.

- `MRG_SPCMAT_YN`: [default: N]

  Defines whether the program produces speciated output.

    - Y: Read the speciation matrix produced by the **Spcmat** program

    - N: Do not read the speciation matrix

- `MRG_TEMPORAL_YN`: [default: N]

  Defines whether the program produces temporally allocated output.

    - Y: Read the temporal matrix produced by the **Temporal** program

    - N: Do not read the temporal matrix

- `MRG_TOTOUT_UNIT`: [default: input file units, or per day if hourly input is used]

  This variable is like the `MRG_GRDOUT_UNIT` variable, except that it controls the state and county totals reports separately from the gridded outputs. The only difference is the temporal resolution of `MRG_TOTOUT_UNIT`, which always outputs the emission values as daily totals, no matter what units are used.

- `SMK_ASCIIELEV_YN`: [default: N]

  Defines whether the program produces an ASCII elevated point sources file, suitable for postprocessing with the Ptsrce program in EPS2.0/EPS2.5.

    - Y: Create the ASCII elevated point sources file

    - N: Do not create the ASCII elevated point sources file

- `SMK_AVEDAY_YN`: See description in Section 6.2.3.2, "Input Environment Variables" [221]

  NOTE: This setting affects the **Smkmerge** results only when hourly emissions are not merged. To use hourly average-day emissions in **Smkmerge**, this variable must be set to Y for the **Temporal** program, so that average-day emissions will be used in creating the hourly emissions.

- `SMK_EMLAYS`: See description in Section 6.6.3.2, "Input Environment Variables" [237]

  Used to set the default value for `UAM_LAYERS` when the **Smkmerge** output is to be used in UAM-V elevated-point-source output processing.

- `SMK_PING_METHOD`: See description in Section 6.3.3.2, "Input Environment Variables" [226]

  Creates CMAQ or MAQSIP PinG file or flags PinG sources in UAM-style elevated ASCII file (when the `SMK_ASCIIELEV_YN` environment variable is set to Y).

- `SMK_REPORT_TIME`: [default: 230000]

  Time for writing daily reports from program (HHMMSS).

- `SMK_SRCGROUP_OUTPUT_YN`: [default: N]

  Indicates if source apportionment emissions should be output

  - Y: Group sources and output source apportionment files

  - N: Do not perform any source apportionment processing

- `SMKMERGE_CUSTOM_OUTPUT`: [default: N]

  Defines his/her own output file names from **Smkmerge**. Manually define environment variables for those output file names below:

  - `[E|A|M|P]OUT` : Processing source output file

  - `ELEV` : Elevated-point-sources file, produced if `SMK_ASCIIELEV_YN` is set to Y; if PinG sources have also been selected, then the stack diameters listed in this file will be set to the negative value of their actual value.

  - `PING` : CMAQ PinG file, produced if `SMK_PING_METHOD` is set to 1

  - `INLN` : CMAQ 2-D surface layer emissions file, produced if `SMK_ELEV_METHOD` is set to 2. For in-line processing plume rise processing, `MRG_LAYERS_YN` must set to N to create hourly speciated 2-D surface layer emissions file for CMAQ version 4.7 or later.

  - `REP[E|A|M|P]G` : Emission total report file, produced if either `MRG_REPSTA_YN` or `MRG_REPCNY_YN` is set to Y

The rest of the variables in this list are relevant only when the `SMK_ASCIIELEV_YN` environment variable is set to Y. The UAM-style elevated-point-source file referenced in the following descriptions is produced by **Smkmerge** and used as input to the UAM preprocessor Ptsrce.

- `UAM_LAYERS`: [default: set by value of `SMK_EMLAYS`]

  Number of layers to output to the elevated-point-source file.

- `UAM_NOTE`: [default: "UAM elev pt emis from Smkmerge"]

Message to appear as a note in the elevated-point-source file.

- UAM_NPARAM: [default: 20]

  Number of parameters for control packet to appear in the elevated-point-source file.

- UAM_PRINT_LOCATIONS: [default: 0]

  Flag printed in the elevated-point-source file to control whether Ptsrce will print the locations table. 0=No; 1=Yes.

- UAM_PRINT_METHODS: [default: 0]

  Flag printed in the elevated-point-source file to control whether Ptsrce will print the methods table. 0=No; 1=Yes.

- UAM_PRINT_OUTGRD: [default: 0]

  Flag printed in the elevated-point-source file to control whether Ptsrce will print the output grid. 0=No; 1=Yes.

- UAM_PRINT_UNITS: [default: 0]

  Flag printed in the elevated-point-source file to control whether Ptsrce will print the units table. 0=No; 1=Yes.

- UAM_PRINT_VALUES: [default: 0]

  Flag printed in the elevated-point-source file to control whether Ptsrce will print the values table. 0=No; 1=Yes.

- UAM_PRINT_VERTMETH: [default: 0]

  Flag printed in the elevated-point-source file to control whether Ptsrce will print the vertical methods table. 0=No; 1=Yes.

- UAM_PTSRCE_OUTUNIT: [default: 20]

  Ptsrce output file unit number in the elevated-point-source file.

- UAM_VERTICAL_METHOD: [default: STACKHGT]

  Vertical method in the elevated-point-source file. Appropriate values are STACKHGT and PLUMERISE.

- UAM4_DIFFBREAK_UNIT: [default: 0]

  DIFFBREAK file unit number in the elevated-point-source file.

- UAM4_HTLOWR: [default: 20]

  Minimum height (in meters) of cells between the surface and the "diffbreak" in the elevated-point-source file.

- UAM4_HTSFC: [default: 0]

  Height of the surface layer (in meters) in the elevated-point-source file.

- `UAM4_HTUPPR`: [default: 100]

  Minimum height (in meters) of cells between the "diffbreak" and the top of the model in the elevated-point-source file.

- `UAM4_LAYABOVE`: [default: `UAM_LAYERS` - `UAM4_LAYBELOW`]

  Number of layers above the "diffbreak" in the elevated-point-source file.

- `UAM4_LAYBELOW`: [default: 3]

  Number of layers below the "diffbreak" in the elevated-point-source file.

- `UAM4_METSCALARS_UNIT`: [default: 15]

  METSCALARS file unit number in the elevated-point-source file. Used only for `UAM_VERTIC-AL_METHOD` set to "PLUMERISE".

- `UAM4_REGIONTOP_UNIT`: [default: 0]

  REGIONTOP file unit number in the elevated-point-source file.

- `UAM4_TEMPERATUR_UNIT`: [default: 14]

  TEMPERATUR file unit number in the elevated-point-source file. Used only for `UAM_VERTIC-AL_METHOD` set to "PLUMERISE".

- `UAM4_WIND_UNIT`: [default: 16]

  WIND file unit number in the elevated-point-source file. Used only for `UAM_VERTICAL_METHOD` set to "PLUMERISE".

6.14.3.3. Output Files

The output files produced by **Smkmerge** are determined by the values of the input environment variables.

File Name	Format	Description
[A|B|M|P|E]G[T][S][C][3D][_S|_L][_O]	I/O API NetCDF	Merged emission file, produced when `MRG_GRDOUT_YN` is set to Y; actual file name is determined by the environment variables used to configure **Smkmerge**, as follows: • [area|biogenic|mobile|point|every] (`MRG_SOURCE`) • T = temporally allocated (`MRG_TEMPOR-AL_YN`) • S = speciated (`MRG_SPCMAT_YN`) • C = controlled (`MRG_CTLMAT_MULT` or `MRG_CTLMAT_REAC`) • 3D = layered (`MRG_LAYERS_YN`) • _S = mass-based speciation (`SPC_INPUT`) • _L = mole-based speciation (`SPC_INPUT`) • _O = used when only merging biogenic sources (`MRG_SOURCE` = B)

File Name	Format	Description
`ELEV[T]S[C][_S|_L]`	ASCII	Elevated-point-sources file, produced if `SMK_ASCIIELEV_YN` is set to Y; if PinG sources have also been selected, then the stack diameters listed in this file will be set to the negative value of their actual value
`LOGFILE`	ASCII	Log generated from executing **Smkmerge**
`PING[T][S][C][_S|_L]`	I/O API NetCDF	CMAQ or MAQSIP PinG file, produced if `SMK_PING_METHOD` is set to 1
`REP[A|B|M|P|E]G[T][S][C][_S|_L]`	ASCII	Emission totals report file, produced if either `MRG_REPSTA_YN` or `MRG_REPCNY_YN` is set to Y
`SGINLNTS[_S|_L]`	I/O API NetCDF	Emissions by source group, formatted as CMAQ in-line emissions data, produced if `SMK_SRCGROUP_OUTPUT_YN` is set to Y
`SRCGROUPS_OUT`	I/O API NetCDF	Source group information, produced if `SMK_SRCGROUP_OUTPUT_YN` is set to Y
`SRCGRP_REPORT`	ASCII	Source group report file, produced if `SMK_SRCGROUP_OUTPUT_YN` is set to Y

6.15. Spcmat

6.15.1. Description

Processing categories: area, mobile, point

An emission inventory is built and reported for a variety of compounds or chemical classes such as CO, NO_x, VOC, PM_{10}, and SO_2. However, AQM photochemical mechanisms (e.g., CB4, RADM2) contain a simplified set of equations that use representative "model species" to represent atmospheric chemistry. Therefore, factors are needed to convert the emissions from the chemical classes in the emission inventory to the species in the intended photochemical mechanism. The **Spcmat** program produces matrices that contain the factors for performing this conversion.

The speciation factors are provided to **Spcmat** using "speciation profiles", which are applied to the inventory's sources using a "speciation cross-reference". For inventories that contain VOC mass, an additional factor is needed that converts the VOC mass to Total Organic Gases (TOG) mass, since the profiles used for VOC are actually computed based on TOG. The TOG profiles are used (instead of VOC profiles) since some of the TOG mass gets assigned to model species. Using VOC-only profiles would underestimate the volatile mass being emitted. Historically in SMOKE, the VOC-to-TOG conversion factors have been assigned by SCC or FIPS and SCC. Starting with SMOKE version 2.3, however, these factors can optionally be assigned by speciation profile ID codes. In fact, since the VOC-to-TOG factors come directly from the raw data used to create speciation profiles factors (such as the SPECIATE4.0 database), the most consistent way to assign these factors is actually by speciation profile ID. SMOKE version 2.3 is backward compatible with existing GSCNV file formats.

Spcmat outputs both mass-based and mole-based speciation factors for all sources. Previous systems have assumed that the average molecular weight of the model species was sufficient to represent the molecular weight of a species emitted by a specific source. In fact, the actual molecular weight of a model species is different for different types of emission processes. To represent the mass values more accurately, the mass and mole factors are input, applied, and output separately by Spcmat.

The speciation cross-reference file input to **Spcmat** specifies the speciation profile to be used for transforming the inventory pollutants to the model species for each source or type of source. **Spcmat** permits profiles to be assigned based on the country, state, and county code, the SCC, and other source characteristics. In the hierarchical system used to apply the profiles, the most specific assignments are applied first, followed by the next most specific, and so on. This system ensures that the most-specific matches override the less-specific ones.

For stationary area/nonpoint sources, the available combinations and their order are the following, starting with the most specific. "Left *x* of SCC" refers to an 8- or 10-digit SCC with all digits except the first *x* set to 0. "Left 2 of SIC code" refers to a 4-digit SIC code with the third and fourth digits set to 0.The maximum field width in SMOKE and its input files for SCC is 20 characters as of SMOKE v4.0. The 8 or 10 digit SCC are still supported, but if the SCC is greater than 10 digits the SCC hierarchial approach will not be supported. The SIC may also be up to 20 characters in length, but if the SIC is larger than 4 characters then the hierarchial approach will not be used. (NOTE: only for SCC less than or equal to 10 characters and SIC less than or equal to 4 characters)

1. Country/state/county code, MACT code, and 8- or 10-digit SCC
2. Country/state/county code and MACT code
3. Country/state code, MACT code, and 8- or 10-digit SCC
4. Country/state code and MACT code
5. MACT code and 8- or 10-digit SCC
6. MACT code
7. Country/state/county code and 4-digit SIC code
8. Country/state/county code and left 2 of SIC code
9. Country/state code and 4-digit SIC code
10. Country/state code and left 2 of SIC code
11. 4-digit SIC code
12. Left 2 of SIC code
13. Country/state/county code and 8- or 10-digit SCC
14. Country/state/county code and left 6 or left 7 of SCC
15. Country/state code and 8- or 10-digit SCC
16. Country/state code and left 6 or left 7 of SCC
17. 8- or 10-digit SCC
18. Left 6 or left 7 of SCC
19. Country/state/county code
20. Country/state code
21. Ultimate default (country/state/county code = 0, SCC = 0)

For nonroad sources, the available combinations and their order are:

1. Country/state/county code and 10-digit SCC
2. Country/state/county code and left 7 of SCC
3. Country/state code and 10-digit SCC
4. Country/state code and left 7 of SCC
5. 10-digit SCC
6. Left 7 of SCC
7. Country/state/county code
8. Country/state code
9. Ultimate default (country/state/county code = 0, SCC = 0)

For point sources, the available combinations depend on the source definition. A header in the speciation cross-reference file format allows you to indicate the source definition. For inventories, a source is defined as the unique combination of country/state/county code, plant ID, point ID, stack ID, segment, and SCC. For point sources, the available combinations and their order are:

1. Country/state/county code, plant ID, point ID, stack ID, segment ID, SCC, and pollutant
2. Country/state/county code, plant ID, point ID, stack ID, segment ID, and pollutant
3. Country/state/county code, plant ID, point ID, stack ID, and pollutant
4. Country/state/county code, plant ID, point ID, and pollutant
5. Country/state/county code, plant ID, SCC, and pollutant
6. Country/state/county code and plant ID
7. Country/state/county code, MACT code, and SCC
8. Country/state/county code and MACT code
9. Country/state code, MACT code, and SCC
10. Country/state code and MACT code
11. MACT code and SCC
12. MACT code
13. Country/state/county code and 4-digit SIC code
14. Country/state/county code and left 2 of SIC code
15. Country/state code and 4-digit SIC code
16. Country/state code and left 2 of SIC code
17. 4-digit SIC code
18. Left 2 of SIC code
19. Country/state/county code and SCC
20. Country/state/county code and left 6 of SCC
21. Country/state/county code and left 3 of SCC
22. Country/state/county code and left 1 of SCC
23. Country/state code and SCC
24. Country/state code and left 6 of SCC
25. Country/state code and left 3 of SCC
26. Country/state code and left 1 of SCC
27. SCC
28. Left 6 of SCC
29. Left 3 of SCC
30. Left 1 of SCC
31. Country/state/county code
32. Country/state code
33. Ultimate default (country/state/county code = 0, SCC = 0)

6.15.2. Processing Order

Smkinven must be run before **Spcmat**.

Spcmat must be executed before speciated emissions can be produced by **Smkmerge**.

6.15.3. Files and Environment Variables

Figure 6.22. Spcmat input and output files
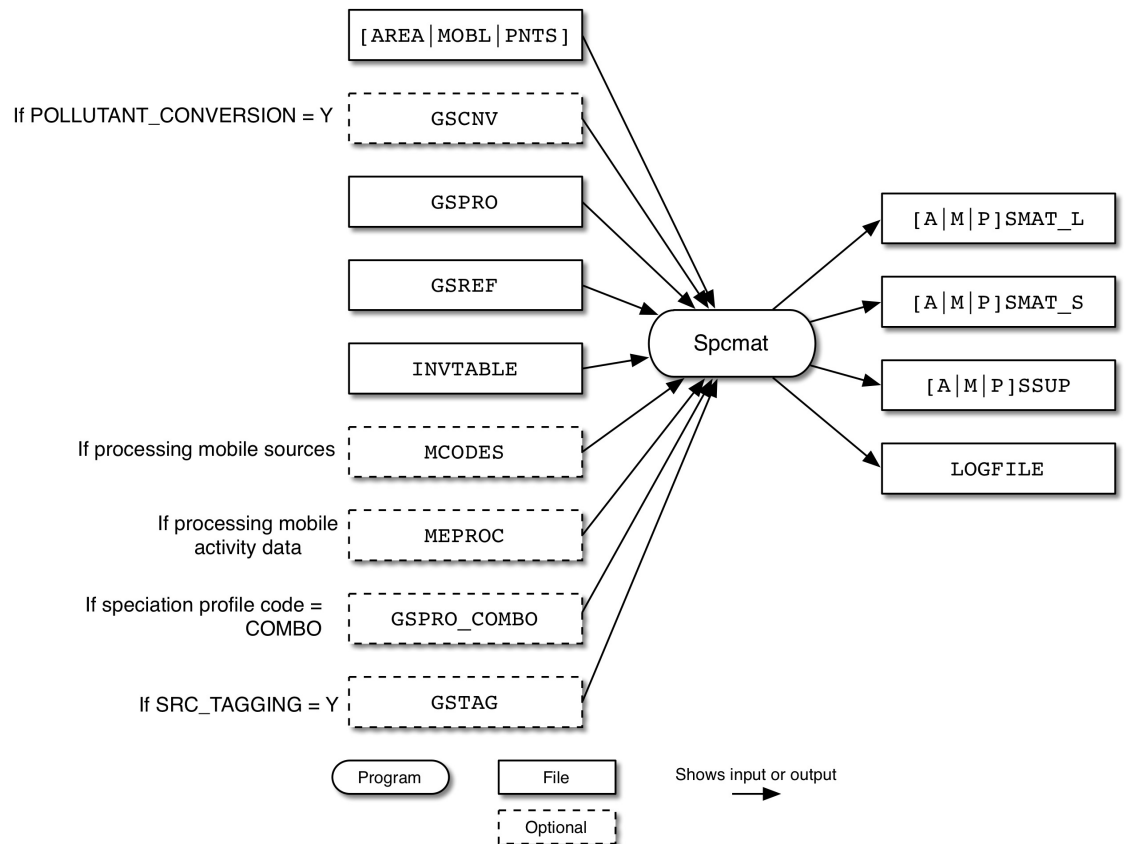
Figure 6.22, "**Spcmat** input and output files" [289] shows the input and output files for the **Spcmat** program. The input files include the inventory file output from **Smkinven** (`AREA`, `MOBL`, or `PNTS`), the optional pollutant conversion file (`GSCNV`), the speciation profiles file (`GSPRO`), the speciation cross-reference file (`GSREF`), the combination speciation profiles file (`GSPRO_COMBO`), and the inventory data table (`INVTABLE`). For mobile sources only, **Spcmat** uses the mobile emission processes file (`MEPROC`). More informaton of `GSPRO_COMBO` file is available at Section 8.5.3, "`GSPRO_COMBO`: Combination speciation profiles file " [391].

Spcmat outputs a mole-based speciation matrix file (`ASMAT_L`, `MSMAT_L`, or `PSMAT_L`) and a mass-based speciation matrix file (`ASMAT_S`, `MSMAT_S`, or `PSMAT_S`). The program also outputs a speciation supplemental file (`ASSUP`, `MSSUP`, or `PSSUP`) and a log file (`LOGFILE`).

6.15.3.1. Input Files

File Name	Format	Description
[AREA|MOBL|PNTS]	SMOKE map	Intermediate inventory file produced by **Smkinven**
GSCNV (optional)	ASCII	Pollutant conversion file, used only if `POLLUTANT_CONVER-SION` = Y.

File Name	Format	Description
GSPRO	ASCII	Speciation profiles
GSREF	ASCII	Speciation cross-reference file
INVTABLE	ASCII	Inventory data table listing activity and pollutant names
MEPROC (optional)	ASCII	Mobile emission processes file

6.15.3.2. Input Environment Variables

- FULLSCC_ONLY: See description in Section 6.2.3.2, "Input Environment Variables" [221]. In this context, this option affects cross-referencing for the speciation cross-reference files by preventing left-7 SCC assignments. The application GSCNV files are not affected by this option when using FIPS/SCC assignments, and will continue to use left-7 SCC assignments even when FULLSCC_ONLY is set to "Y".

- NO_SPC_ZERO_EMIS: [default: N]

  Defines whether the program applies the speciation profiles to zero emission sources.

  - Y: Skip applying speciation profile to zero emission source

  - N: Do not skip

- POLLUTANT_CONVERSION: [default: N]

  Defines whether the program uses the pollutant-to-pollutant conversion file (Section 8.5.1, "GSCNV: Pollutant-to-pollutant conversion file" [389]).

  - Y: Use conversion file

  - N: Do not use conversion file

- REPORT_DEFAULTS: See description in Section 6.2.3.2, "Input Environment Variables" [221]

- SMK_SOURCE: See description in Section 6.2.3.2, "Input Environment Variables" [221]

- SPCMAT_PERIOD: [default:1]

  Defines period number which could be month. It must match to the period number in (Section 8.5.3, "GSPRO_COMBO: Combination speciation profiles file " [391]).

- SRC_TAGGING: [default: N]

  Defines whether the program uses the speciation tagging cross-reference file or not (Section 8.5.5, "GSTAG: Speciation tagging cross-reference file " [394]).

  - Y: Use speciation tagging cross-reference file

  - N: Do not use speciation tagging cross-reference file

- DEFAULT_CONV_FAC_ERROR: [default: N]

  Defines whether the program treats as an error when there is no matched VOC to TOG conversion factor available in the pollutant-to-pollutant conversion file (Section 8.5.1, "GSCNV: Pollutant-to-pollutant conversion file" [389]).

- Y: Use speciation tagging cross-reference file

- N: Do not use speciation tagging cross-reference file

6.15.3.3. Output Files

File Name	Format	Description
`[A|M|P]SMAT_L`	I/O API NetCDF	Mole-based speciation matrix
`[A|M|P]SMAT_S`	I/O API NetCDF	Mass-based speciation matrix
`[A|M|P]SSUP`	I/O API NetCDF	Speciation supplemental file
`LOGFILE`	ASCII	Log generated from executing **Spcmat**

6.16. Temporal

6.16.1. Description

Processing categories: area, mobile, point

The **Temporal** program produces a file with hourly inventory pollutant emissions. Unlike the SMOKE matrices produced by **Cntlmat**, **Grdmat**, and **Spcmat**, the output file from **Temporal** contains the actual emissions data. For all source categories, it uses temporal profiles, temporal cross-references, and the time zone of each source to create the hourly emissions output file. It can also read in day-specific and hour-specific inventory data in SMOKE format, and it uses the most detailed data it can when multiple types of data are available for the same source. For example, if a source had annual, day-specific, and hour-specific data, **Temporal** would use the hour-specific emissions. The program can also apply source-specific hourly profiles from the hour-specific data file, and it can override all temporal profiles and use temporally uniform profiles for all sources.

Temporal treats the temporal profiles as *local* profiles. In other words, the profile applied to the source is adjusted based on the difference between the time zone of the source (determined by the COSTCY file) and the output time zone (determined by the OUTZONE environment variable). SMOKE automatically considers Daylight Saving time when converting from a region's standard time zone to the output time zone, and SMOKE can exclude regions that never use Daylight Saving time based on the COSTCY file. Note that **Temporal** has been tested for time zones in the Western Hemisphere, but we have not tested it for time zones in the Eastern Hemisphere.

Within the **Temporal** program, the processing occurs as follows:

- Temporal profiles are assigned for all sources.

- Monthly (year-to-month) temporal profiles can be applied.

- Daily (month-to-day) temporal profiles can be applied.

- Weekly (month-to-weekly) temporal profiles can be applied.

- Day-of-week specific hourly diurnal temporal profiles (Monday,,,Sunday, Weekday, Weekend, or All days) are applied.

- Date-specific hourly diurnal temporal profiles (year-to-hour, month-to-hour or day-to-hour) can be applied.

- Day-specific emissions or activities are read in and used to overwrite daily emissions for the appropriate sources

- Hour-specific emissions or activities are read in and used to overwrite hourly emissions for the appropriate sources.

Hourly temporal profiles can be applied for weekdays and weekend days by default in the system. By breaking up the processing into different periods, you can apply different temporal profiles for different days of the week. A future version of SMOKE will permit different hourly profiles for every day of the week in a single run of the **Temporal** program.

A special holidays file should be provided so that holidays can receive special treatment. This file specifies which daily profiles should be applied to holidays.

A many-step hierarchy is used in applying the temporal cross-referencing. SMOKE preferentially matches each source to the most specific line in the cross-reference file. When there are missing entries in the cross-reference file, **Temporal** first looks for cross-reference entries for other species at the same level in the search hierarchy before descending to the next level in the hierarchy. For example, if a source has a temporal cross-reference entry for CO but not for NO_x, **Temporal** uses the source-specific CO temporal cross reference entry for allocating NO_x from that source in preference to using a default NO_x temporal profile.

For area sources, the available combinations are the following, starting with the most specific. "7-digit SCC" means a 10-digit SCC with the last three digits set to zero. The maximum field width in SMOKE and its input files for SCC is 20 characters as of SMOKE v4.0. The 8 or 10 digit SCC are still supported, but if the SCC is greater than 10 digits the SCC hierarchial approach will not be supported. The SIC may also be up to 20 characters in length, but if the SIC is larger than 4 characters then the hierarchial approach will not be used. (NOTE: only for SCC less than or equal to 10 characters and SIC less than or equal to 4 characters)

1. Country/state/county code, 10-digit SCC, and pollutant
2. Country/state/county code, 7-digit SCC, and pollutant
3. Country/state code, 10-digit SCC, and pollutant
4. Country/state code, 7-digit SCC, and pollutant
5. 10-digit SCC and pollutant
6. 7-digit SCC and pollutant
7. Country/state/county code and 10-digit SCC
8. Country/state/county code and 7-digit SCC
9. Country/state code and 10-digit SCC
10. Country/state code and 7-digit SCC
11. 10-digit SCC
12. 7-digit SCC
13. Country/state/county code
14. Country/state code
15. Ultimate default (no pollutant, country/state/county code = 0, SCC = 0)

For point sources, the available combinations depend on the source definition. A header is used in the file to indicate the source definition. For inventories, a source is defined as the unique combination of country/state/county code, plant ID, point ID, stack ID, segment, and SCC. "5-digit SCC" means a 10-digit SCC with the last 5 digits set to zero. If only 8-digit SCCs are available, these can be used and SMOKE will assume leading zeros. In this case, the "5-digit SCC" becomes a "3-digit SCC". The maximum field width in SMOKE and its input files for SCC is 20 characters as of SMOKE v4.0. The 8 or 10 digit SCC are still supported, but if the SCC is greater than 10 digits the SCC hierarchial approach will not be supported. The SIC may also be up to 20 characters in length, but if the SIC is larger than 4 characters then the hierarchial approach will not be used.

The hierarchy of cross-referencing is the following. (NOTE: only for SCC less than or equal to 10 characters, SIC less than or equal to 4 characters)

1. Country/state/county code, 10-digit SCC, plant ID, point ID, stack ID, segment, and pollutant
2. Country/state/county code, 10-digit SCC, plant ID, point ID, stack ID, and pollutant
3. Country/state/county code, 10-digit SCC, plant ID, point ID, and pollutant
4. Country/state/county code, 10-digit SCC, plant ID, and pollutant
5. Country/state/county code, 10-digit SCC, plant ID, point ID, stack ID, and segment
6. Country/state/county code, 10-digit SCC, plant ID, point ID, and stack ID
7. Country/state/county code, 10-digit SCC, plant ID, and point ID
8. Country/state/county code, 10-digit SCC, and plant ID
9. Country/state/county code, and plant ID
10. Country/state/county code, 10-digit SCC, and pollutant
11. Country/state/county code, 5-digit SCC, and pollutant
12. Country/state code, 10-digit SCC, and pollutant
13. Country/state code, 5-digit SCC, and pollutant
14. 10-digit SCC and pollutant
15. 5-digit SCC and pollutant
16. Country/state/county code and 10-digit SCC
17. Country/state/county code and 5-digit SCC
18. Country/state code and 10-digit SCC
19. Country/state code and 5-digit SCC
20. 10-digit SCC
21. 5-digit SCC
22. Country/state/county code
23. Country/state code
24. Ultimate default (all values = 0)

6.16.2. Processing Order

Smkinven must be run before **Temporal**.

Temporal must be run before gridded, hour-specific emissions can be produced using the **Smkmerge** program.

6.16.3. Files and Environment Variables

Figure 6.23. Temporal input and output files for area sources
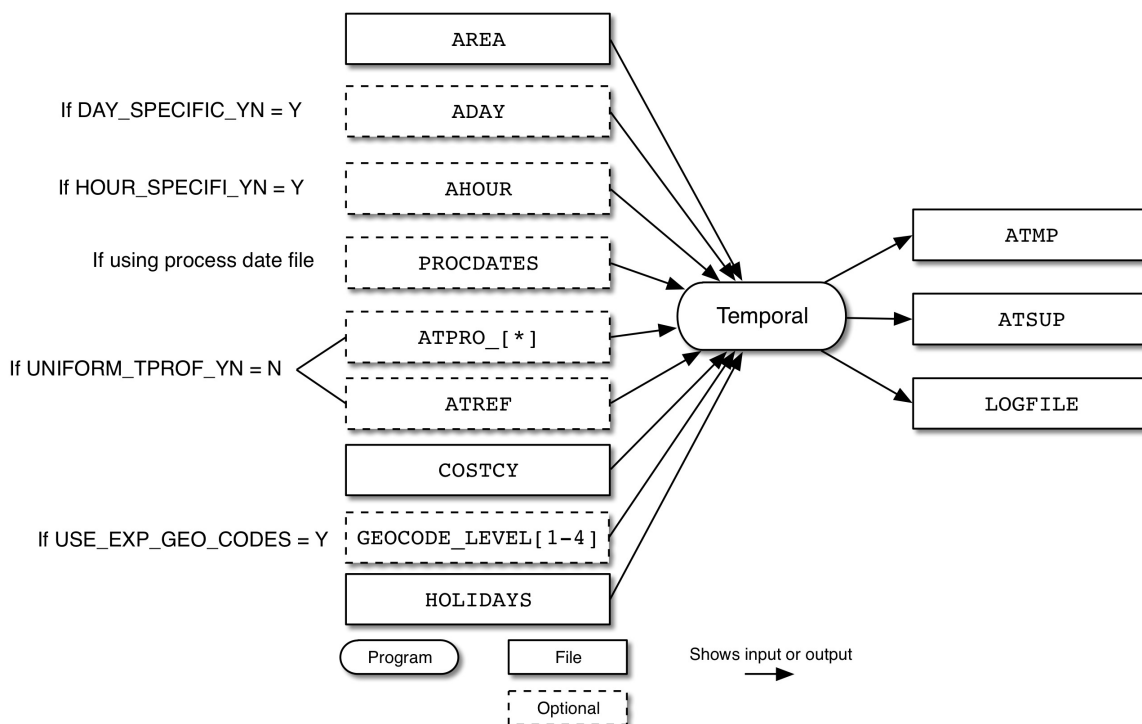
Figure 6.23, "**Temporal** input and output files for area sources" [294] shows the input and output files for the **Temporal** program for area sources. The input files include the inventory file output from **Smkinven** (AREA), the temporal profiles file (ATPRO), the temporal cross-reference file (ATREF), the country, state, and county codes file (COSTCY), the holidays file HOLIDAYS), and optional process date file (PROCD-ATES).

Temporal outputs an hourly emissions file (ATMP), a temporal allocation supplemental file (ATSUP), and a log file (LOGFILE).

Figure 6.24. Temporal input and output files for mobile sources
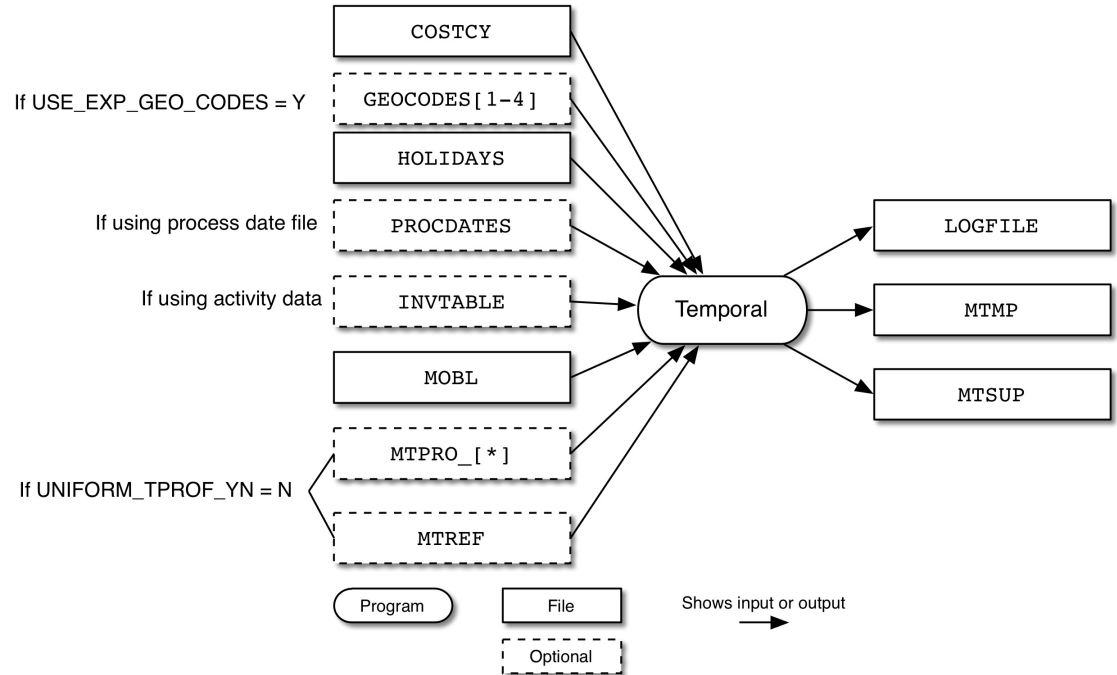
Figure 6.24, "**Temporal** input and output files for mobile sources" [295] shows the input and output files for the **Temporal** program for mobile sources. The input files include the country, state, and county codes file (COSTCY), the holidays file (HOLIDAYS), the optional process date file (PROCDATES) and, if activity data are used, the inventory data table (INVTABLE). **Temporal** also uses the inventory file output from **Smkinven** (MOBL), the temporal profiles file (MTPRO), and the temporal cross-reference file (MTREF).

Temporal outputs a log file (LOGFILE), an hourly emissions file (MTMP), and a temporal allocation supplemental file (MTSUP).

Figure 6.25. Temporal input and output files for point sources
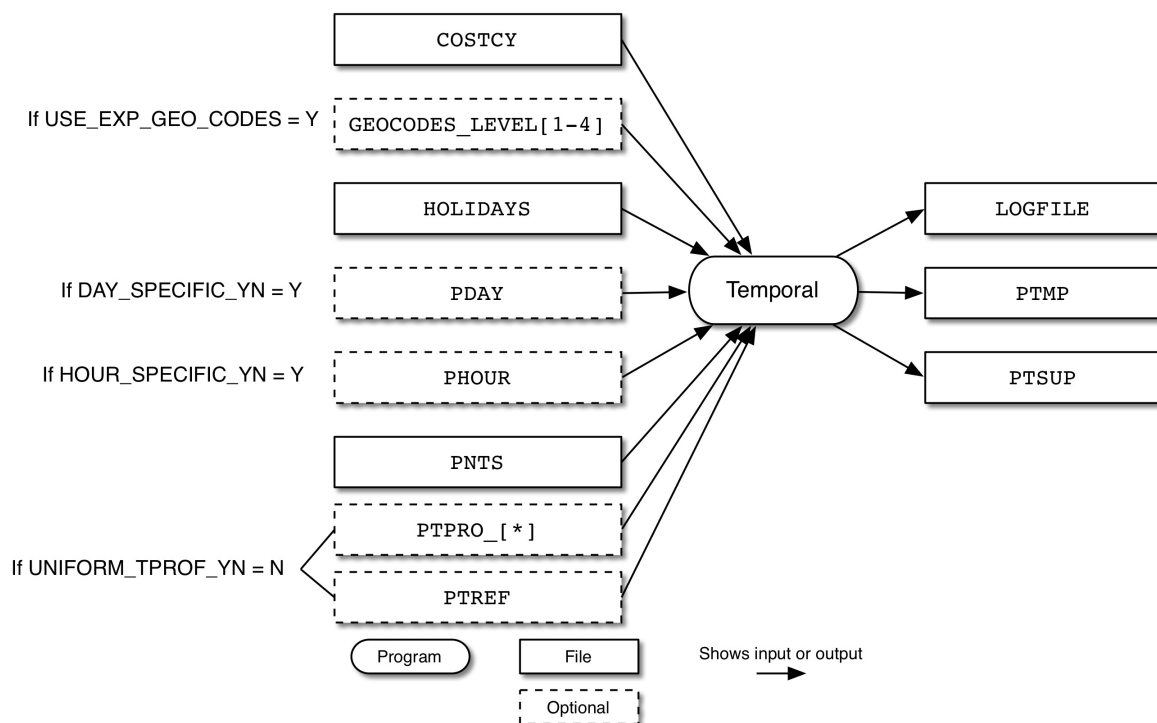
Figure 6.25, "**Temporal** input and output files for point sources" [296] shows the input and output files for the **Temporal** program for point sources. The input files include the country, state, and county codes file (COSTCY), the holidays file (HOLIDAYS) and the optional process date file (PROCDATES). If using day-specific data, the PDAY file is read, and the PHOUR file is read when using hour-specific data. **Temporal** also uses the inventory file output from **Smkinven** (PNTS), the temporal profiles file (PTPRO), and the temporal cross-reference file (PTREF).

Temporal outputs a log file (LOGFILE), an hourly emissions file (PTMP), and a temporal allocation supplemental file (PTSUP).

Optionally, **Temporal** may be run using PROCDATES. This enables the processing of sequential or non-sequential user specified days. If PROCDATES is not assigned, the defaults apply. When running **Temporal** interactively, the following inputs will be requested. The default values for these are set by the G_STDATE, G_STTIME, and G_RUNLEN environment variables.

- Start date: Starting date of simulation in Julian (YYYYDDD) or Gregorian (YYYYMMDD) formats

- Start Time: Starting time of simulation in HHMMSS

- Duration: Simulation duration in hours

PROCDATES: See description in Section 8.3.4, "PROCDATES: Procdates date list" [386].

6.16.3.1. Input Files

File Name	Format	Description
[AREA|MOBL|PNTS]	SMOKE map	Intermediate inventory file produced by **Smkinven**

File Name	Format	Description
`[A|M|P]TPRO_MONTHLY`	ASCII	Monthly Temporal profiles
`[A|M|P]TPRO_DAILY`	ASCII	Month-to-day Temporal profiles
`[A|M|P]TPRO_WEEKLY`	ASCII	Weekly Temporal profiles
`[A|M|P]TPRO_HOURLY`	ASCII	Day-to-hour Temporal profiles
`[A|M|P]TPRO_HOURLY_NCF`	ASCII	Meteorological-based hourly Temporal profiles
`[A|M|P]TREF`	ASCII	Temporal cross-reference file
`COSTCY`	ASCII	Country/state/county codes file, used to determine counties that do not use Daylight Saving time.
`GEO-CODE_LEVEL[1-4]`	ASCII	Geographical level codes and descriptions for gridded inventories, only when USE_EXP_GEO_CODES is set to Y, (COSTCY is ignored when `GEOCODE_LEVEL[1-4]` files are used)
`HOLIDAYS`	ASCII	Holidays file by Gregorian date, for applying holiday temporal profiles.
`PROCDATES` (optional)	ASCII	Processes user specified Gregorian dates
`INVTABLE` (optional)	ASCII	Inventory data table listing pollutant and activity names; required only if activity data are used
`PDAY` (optional)	I/O API NetCDF	Day-specific intermediate inventory file created by **Smkinven** (required if `DAY_SPECIFIC_YN` is set to Y)
`PHOUR` (optional)	I/O API NetCDF	Hour-specific intermediate inventory file created by **Smkinven** (required if `HOUR_SPECIFIC_YN` is set to Y)

6.16.3.2. Input Environment Variables

- `DAY_SPECIFIC_YN`: See description in Section 6.13.3.2, "Input Environment Variables" [265]

- `FULLSCC_ONLY`: See description in Section 6.2.3.2, "Input Environment Variables" [221]. In this context, this option affects cross-referencing for the temporal cross-reference files by preventing left-7 SCC assignments.

- `USE_EXP_GEO_CODES`: [default: N]

  If set to Y then use GEOCODE_LEVEL[1-4] files instead of the COSTCY file to define expanded geographical detail such as country-level(GEOCODE_LEVEL1), country/state-level (GEO-CODE_LEVEL2), country/state/county-level (GEOCODE_LEVEL3) or country/state/county/tribal-level (GEOCODE_LEVEL4).

- `HOUR_SPECIFIC_YN`: See description in Section 6.13.3.2, "Input Environment Variables" [265]

- `HOURLY_TPROF_BASE`: [default: blank]

  Defines the base of hourly temporal profiles. This flag is only applicable when hourly temporal profiles generated by **Gentpro** is available for **Temporal** to use. For a proper processing of hourly temporal profiles from **Gentpro**, user must let **Temporal** know the basis of hourly temporal profiles (i.e., hour of day, hour of month or hour of year). **Gentpro** outputs hourly temporal profiles IDs into temporal cross-reference input file (`[A\|M\|P]TREF`). See detail information at Section 8.3.2, "`[A\|M\|P]TREF`: Temporal cross-reference file" [383]. If `HOUR_PROFID` column is filled by **Gentpro**, user must define

the base of hourly profiles to correctly convert annual/monthly/daily inventories. If processing monthly inventory, user must set it to 'MONTH'.

- YEAR: Convert raw annual inventory to hourly inventory values.

- MONTH: Convert calculated monthly inventory to hourly inventory values.

- OUTZONE: : [default: 0]

  Indicates offset from Greenwich Mean Time (GMT) for episode start date and time. OUTZONE should be set to the standard time for the time zone of interest; SMOKE will adjust to Daylight Saving time for episodes that are during periods and in regions where Daylight Saving time applies.

  Table 2.13, "Example OUTZONE settings and their associated time zones" [67] provides allowed values for OUTZONE. The CMAQ and MAQSIP models expect OUTZONE to be 0, while other models such as UAM-IV, UAM-V, REMSAD, and CAM$_X$ often use time zone 5. All models expect that the emissions and the meteorology data provided to the model will have a consistent time zone. SMOKE expects that the meteorology data will be in GMT.

  Note that SMOKE expects OUTZONE to be set as a positive number for time zones in the Western Hemisphere, although standard notation would list these as negative values. For example, Eastern Standard Time is -5:00 hours from GMT, but OUTZONE for EST in SMOKE is 5.

- RENORM_TPROF: [default: Y]

  Defines whether **Temporal** normalizes the temporal profiles or uses them as they exist in the data file. If the profiles are specified as decimal values and sum to 1, then specify N. Otherwise, specify Y and the program will sum the values associated with each profile and then divide each value by the sum, thereby creating a normalized profile.

  - Y: Normalize the temporal profiles

  - N: Use the temporal profile values as they exist in the data file

- REPORT_DEFAULTS: See description in Section 6.2.3.2, "Input Environment Variables" [221]

- SMK_AVEDAY_YN: See description in Section 6.2.3.2, "Input Environment Variables" [221]

- SMK_EF_MODEL: [default:blank]

  Name of the emission factor model to use. NOTE: User must set it to MOVES when processing activity inventory data (i.e., VMT, VPOP and SPEED).

- SMK_MAXERROR: See description in Section 6.13.3.2, "Input Environment Variables" [265]

- SMK_MAXWARNING: See description in Section 6.13.3.2, "Input Environment Variables" [265]

- SMK_SOURCE: See description in Section 6.2.3.2, "Input Environment Variables" [221]

- UNIFORM_TPROF_YN: [default: N]

  Defines whether to use uniform temporal profiles for all sources.

  - Y: Use uniform temporal profiles

  - N: Use temporal profiles file and temporal cross-reference file

- `ZONE4WM`: [default: Y]

  Defines whether **Temporal** applies weekly and monthly temporal profiles using time zones. If SMOKE does not apply the profiles using the sources' time zones, then all sources change from one day to the next or one month to the next at the same hour. This is what other emission processing systems have typically done, and it is a minor distinction in most cases.

  - Y: Use time zones

  - N: Do not use time zones

6.16.3.3. Output Files

As of SMOKE version 2.3, two new environmental variables `[A|M|P]TMPNAME` and `[A|M|P]TSUPNAME` are required to set the directory and file prefix used to dynamically name the output files `[A|M|P]TMP` and `[A|M|P]TSUP`.

File Name	Format	Description
`[A|M|P]TMP`	I/O API NetCDF	Hourly emissions by source; if using elevated point sources as part of point-source processing, this file contains only the layer-1 emissions
`[A|M|P]TSUP`	ASCII	Temporal allocation supplemental file
`LOGFILE`	ASCII	Log generated from executing **Temporal**

6.17. Tmpbeis3

6.17.1. Description

Processing categories: biogenic

Tmpbeis3 uses version 3.14 or version 3.60 of the BEIS3 model. **Tmpbeis3** reads the gridded, normalized emissions file `B3GRD` and meteorological data from `MET_FILE1` and optionally `MET_FILE2` (these files may be either `MET_CRO_2D` or `MET_CRO_3D`). Note that **Tmpbeis3** uses temperature and surface pressure data that must be in `MET_FILE1`, and radiation/cloud data that may be in `MET_FILE1` or `MET_FILE2`. Within **Tmpbeis3**, speciation profiles are used to speciate biogenic pollutants for appropriate chemical mechanisms. Various speciation profiles for the CB4, RADM2, and SAPRC99 mechanisms are available for BEIS3 modeling. The normalized emissions and meteorological data are used to produce gridded, speciated, hourly biogenic emissions. Additionally, an optional seasonal switch file, `BIOSEASON`, can be used to determine when and where to use summer or winter normalized emissions data on a daily basis.

6.17.2. Processing Order

The **Normbeis3** program must be executed before **Tmpbeis3**.

If you are using the seasonal switch option, the seasonal switch file (`BIOSEASON`) must be created using the utility program **Metscan** before running **Tmpbeis3**.

Tmpbeis3 must be executed before biogenic emissions can be used in **Smkmerge**.

6.17.3. Files and Environment Variables

Figure 6.26. Tmpbeis3 input and output files
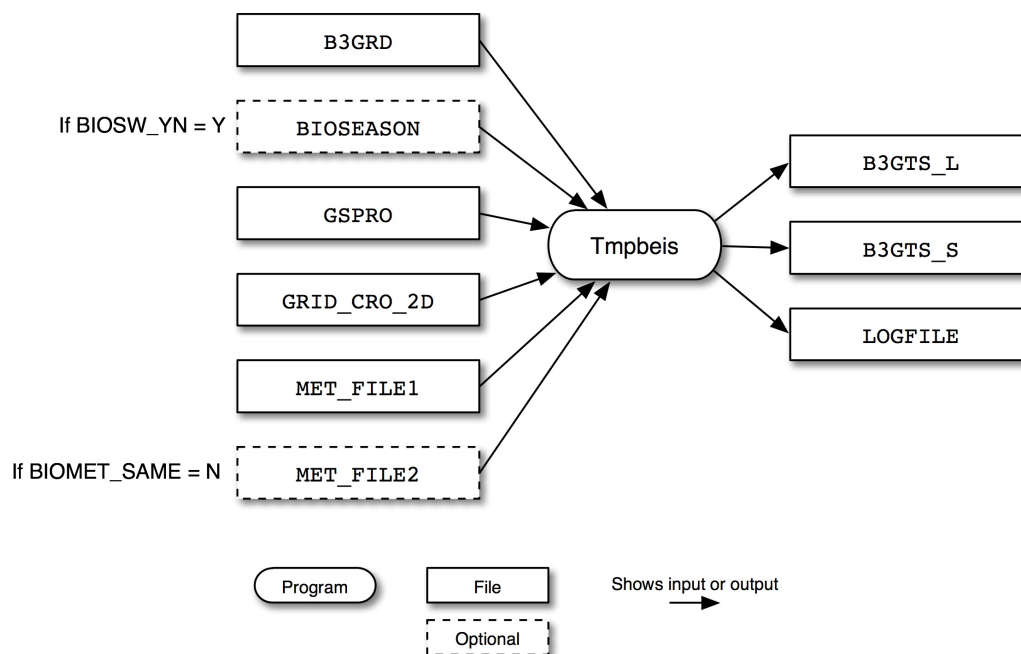
Figure 6.26, "**Tmpbeis3** input and output files" [300] shows the input and output files for the **Tmpbeis3** program. The input files include the gridded, normalized biogenic emissions from **Normbeis3** (B3GRD) and, if BIOSW_YN = Y, then a gridded, seasonal switch file output from **Metscan** is required (BIOSEASON). The biogenic speciation profiles are read from the GSPRO file and the GRID_CRO_2D contains the latitude and longitute for each grid cell. **Tmpbeis3** reads surface pressure and radiation/cloud data from MET_FILE1 and optionally from MET_FILE2 if BIOMET_SAME = N.

Tmpbeis3 outputs gridded, speciated, hourly biogenic emissions files in moles/hour (B3GTS_L) and tons/hour (B3GTS_S). The program also outputs an execution log file (LOGFILE).

6.17.3.1. Input Files

File Name	Format	Description
B3GRD	I/O API NetCDF	Gridded, normalized biogenic emissions, output from **Normbeis3**
BIOSEASON (optional)	I/O API NetCDF	Gridded seasonal switch file containing daily data, with "0" meaning use winter normalized emissions and "1" meaning use the summer normalized emissions, produced by **Metscan**
GSPRO	ASCII	Speciation profiles file (**use "BV309" for CB04 speciation profile or "B10C5" for CB05 speciation profile.**)
GRID_CRO_2D	I/O API NetCDF	Meteorology file containing latitude and longitude coordinates for each grid cell
MET_FILE1	I/O API NetCDF	Meteorology file containing temperature and surface pressure data and possibly radiation/cloud data as well (see BIO-MET_SAME environment variable)

File Name	Format	Description
`MET_FILE2` (optional)	I/O API NetCDF	Meteorology file containing radiation/cloud data if not in `MET_FILE1`
`SOILINP` (optional)	I/O API NetCDF	Gridded soil restart file (use if `INITIAL_RUN = N`)

6.17.3.2. Input Environment Variables

- `BEIS_VERSION`: [default: 3.60]

  Specifies which version of BEIS3 to use.

  - 3.60: Use BEIS version 3.60.

  - 3.14: Use BEIS version 3.14.

- `BG_CLOUD_TYPE`: [default: 1]

  Specifies which meteorological data to use.

  - 1: Use RGND or RSD (solar radiation reaching the ground) from MM5 to calculate PAR. No other options are supported at this time.

- `BIOG_SPRO`: [default: BV310]

  Specifies the speciation profile for biogenic emissions from GSPRO.

  - B3V10: Use BEIS version 3.60.

  - BV313: Use BEIS version 3.14.

- `BIOMET_SAME`: [default: N]

  Defines whether the radiation/cloud data are in the same file as the temperature data (`MET_FILE1`) or in a different file (`MET_FILE2`).

- `BIOSW_YN`: [default: Y]

  Indicates whether to use a gridded seasonal switch file.

  - Y: Use gridded seasonal switch file `BIOSEASON`

  - N: Use environment variable `SUMMER_YN` to determine season

- `INITIAL_RUN`: [default: N]

  Indicates whether a particular day is the first day in a sequence of simulations.

  - Y: Use for the first day in a sequence of simulations. Use if the `SOILINP` file does not yet exist (i.e., the soil NO calculations will not depend on the previous day's rainfall).

  - N: Use for subsequent days in a sequence of simulations. Use if the `SOILINP` file exists.

  Note that BEIS3 will create a `SOILINP` file for days subsequent to the first day in a sequence of simulations.

- `ISLTYP_VAR`: [default: SLTYP]

Specifies the soil type variable name used if `PX_VERSION = Y.`

- `LAI_VAR`: [default: LAI]

  Specifies the variable name for the leaf area index from the me model (e.g. WRF) used for the leaf temperature algorithm (Only for BEIS version 3.60)

- `OUTZONE`: See description in Section 6.16.3.2, "Input Environment Variables" [297]

- `OUT_UNITS`: [default: 1]

  Specifies units for the output variables.

  - 1: moles per hour.

  - 2: moles per second.

- `PRES_VAR`: [default: PRSFC]

  Specifies the variable name for surface pressure data to extract from `MET_FILE1`.

- `PX_VERSION`: [default: N]

  Indicates whether the Pleim-Xiu land surface model was used in the MM5 simulations.

  - Y: Use if the Pleim-Xiu land surface model was used in the MM5 simulations. Indicates that `MET_FILE1` contains soil temperature, soil moisture and soil type fields that can be used, along with rainfall data, for NO soil emission calculations in **Tmpbeis3**.

  - N: Use if the Pleim-Xiu land surface model was not used in the MM5 simulations. Indicates that **Tmpbeis3** will calculate NO soil emissions using only rainfall data.

- `Q2_VAR`: [default: Q2]

  Specifies the variable name for the two meter water vapor mixing ratio (kg/kg) used for the leaf temperature algorithm (Only for BEIS version 3.60).

- `RAD_VAR`: [default: RGRND]

  Specifies the variable name for radiation/cloud data to extract from either `MET_FILE1` or optional `MET_FILE2`.

- `RADYNI_VAR`: [default: RADYNI]

  Specifies the variable name for the aerodynamic resistance variable (s/m) used for the leaf temperature algorithm (Only for BEIS version 3.60).

- `RGRND_VAR`: [default: RGRND]

  Specifies the variable name for the solar radiation reaching the ground used for the leaf temperature algorithm (Only for BEIS version 3.60).

- `RSTOMI_VAR`: [default: RSTOMI]

  Specifies the variable name for the inverse of the stomatal resistance variable (m/s) used for the leaf temperature algorithm (Only for BEIS version 3.60).

- `RC_VAR`: [default: RC]

  Specifies the convective rainfall variable name used in `MET_FILE1`.

- `RN_VAR`: [default: RN]

  Specifies the non-convective rainfall variable name used in `MET_FILE1`.

- `SOILM_VAR`: [default: SOIM1]

  Specifies the soil moisture variable name used if `PX_VERSION = Y`.

- `SOILT_VAR`: [default: SOIT1]

  Specifies the soil temperature variable name used if `PX_VERSION = Y`.

- `SUMMER_YN`: [default: Y]

  Indicates whether summer or winter normalized emissions should be used; only used when `BIOSW_YN` is N.

  - Y: Use summer normalized emissions

  - N: Use winter normalized emissions

- `TEMPG_VAR`: [default: TEMPG]

  Specifies the skin temperature at the ground level (K) used for the leaf temperature algorithm (Only for BEIS version 3.60).

- `TMPR_VAR`: [default: TA]

  Specifies the variable name for temperature to extract from `MET_FILE1`.

- `USTAR_VAR`: [default: USTAR]

  Specifies the variable name for the surface friction velocity (m/s) used for the leaf temperature algorithm (Only for BEIS version 3.60).

6.17.3.3. Output Files

File Name	Format	Description
`B3GTS_L`	I/O API NetCDF	Gridded, speciated, hourly biogenic emissions, in moles/hour
`B3GTS_S`	I/O API NetCDF	Gridded, speciated, hourly biogenic emissions, in tons/hour
`SOILOUT`	I/O API NetCDF	Gridded soil restart file
`LOGFILE`	ASCII	Log generated from executing **Tmpbeis3**

6.17.3.4. Processing Multiple Days with BEIS3

The user must point to the `SOILOUT` file from one day's simulation as the `SOILINP` file for the next day. The user must also decide whether to write over `SOILOUT` files from previous days or create a uniquely named `SOILOUT` file for each day. The latter approach is recommended if the user wishes to retain the capability to restart simulations in the middle of a sequence of simulations.

Chapter 7. SMOKE Quality Assurance

Table of Contents

7.1. Introduction

This chapter provides documentation on SMOKE's quality assurance (QA) capabilities. Quality assurance of emissions modeling results is a very important component of using SMOKE. Because emission modeling is a complex procedure that includes many disparate data sets and many processing steps, it is critical that quality assurance be performed on the emissions modeling results prior to using the model-ready emissions in air quality modeling. Failure to quality assure the results will almost surely result in undesirable errors in SMOKE's output caused by factors such as user inexperience, file formatting errors, misunderstanding of software capabilities, and software bugs. It is far preferable to catch emissions modeling problems during emissions modeling rather than after the emissions results have already been used and policy decisions made based on them.

SMOKE quality assurance consists of at least the following components:

- Setting up SMOKE using available documentation (this manual) on file formats and script setup.

- Reviewing log files for errors and warnings, then resolving any problems.

- Reviewing reports generated by SMOKE core programs (e.g., reports from the **Cntlmat** program indicating which controls were applied and the changes made to the inventory emissions).

- Reviewing and customizing reports generated by the **Smkreport** program, and comparing these to one another and to totals from the emission inventory.

In this chapter, we provide documentation on the **Smkreport** program (Section 7.2, "**Smkreport** Program" [305]), on its main input file `REPCONFIG` (Section 7.3, "`REPCONFIG` Input File" [308]), and on the **Smkreport** output reports (Section 7.4, "Reports Created by **Smkreport**" [326]). In the final section of this chapter (Section 7.5, "Quality Assurance Methods" [335]), we outline QA methods that SMOKE users can apply to ensure that their inventory and emissions modeling efforts are as correct as possible.

7.2. Smkreport Program

7.2.1. Description

Smkreport uses the SMOKE intermediate files generated by other SMOKE programs to create a large variety of emissions and activity data totals to be used for QA of the emissions data. This QA is in addition to the QA performed by other SMOKE programs for checking file formats and input quality, and it focuses on analysis of the emissions values processed and output by SMOKE. **Smkreport** operates on one source category (area, mobile, or point) in a given run, though it does not operate on biogenic emissions because

biogenic does not have the same structure of intermediate files that **Smkreport** uses for inputs. Each run of **Smkreport** is also limited to one speciation scenario, one grid projection and resolution, and one hourly emissions file. **Smkreport** can also read in growth or multiplicative control matrices, but not reactivity control matrices. All of these component datasets are optional, however, and their use is controlled by the settings in the Report Configuration File (REPCONFIG), described in Section 7.3, "REPCONFIG Input File" [308]. Analysis of multiple source categories (area/mobile/point) in the same report is a postprocessing step in which the user can import multiple reports from different source categories and analyze across multiple source categories.

The most important input file to **Smkreport** is the REPCONFIG file. Through this file, you have a tremendous amount of flexibility in generating different types of reports. Options are invoked by using reporting "instructions", which you prepares before using **Smkreport**. See the REPCONFIG documentation in Section 7.3, "REPCONFIG Input File" [308] for more information on the reporting options. Additionally, in Section 7.5, "Quality Assurance Methods" [335] we provide some guidance on the types of analyses you can perform using reports created by **Smkreport**, and reasons why you might want to use those analyses.

Smkreport can generate multiple reports with a single run of the program. Multiple report writing is helpful, because a large part of the time that the program takes to run is reading in the SMOKE intermediate files. Note that the multiple reports are all for a single source category for one run of **Smkreport**. As described in the Section 7.4, "Reports Created by **Smkreport**" [326], the report output names are configured by the REPCONFIG file. You may specify either physical file names (including the path) in REPCONFIG, or you may specify logical file names and then make sure your script sets the logical file names to physical file names. Chapter 4, *Using SMOKE Scripts* [117] describes the default configuration of reports in the example SMOKE scripts. At the end of Chapter 4, *Using SMOKE Scripts* [117], we also provide information about how to configure the scripts to change the output reports that SMOKE will create.

7.2.2. Processing Order

Smkreport can only be used after *at least* **Smkinven** (for inventory import) has been run. Depending on the reporting options selected in the REPCONFIG input file, other processing steps may need to have been run. For example, to create a report of county total emissions after gridding, the **Grdmat** (for spatial allocation) program must have been run already to create the gridding matrix. **Smkreport** can use output files from the following programs: **Smkinven**, **Temporal** (for temporal allocation), **Spcmat** (for chemical speciation), **Grdmat**, **Elevpoint** (for elevated source selection), **Laypoint** (for plume rise calculations), and **Cntlmat**.

7.2.3. Files and Environment Variables

7.2.3.1. Input Files

The input files depend upon the SMK_SOURCE input instruction in the REPCONFIG input file setting when the program is started. With the exception of the SMOKE intermediate inventory file (AREA, MOBL, PNTS), whether a file is optional or not depends on the settings in REPCONFIG. If **Smkreport** is being executed to produce speciated, gridded, temporally allocated area sources, for instance, then the area-source inventory, speciation matrix, gridding matrix, and hourly emissions are required. Also, the source category of the input file depends on the REPCONFIG file setting for SMK_SOURCE. For example, if only area sources are being merged (SMK_SOURCE is set to A), then only area-source-related inputs are required.

File Name	Format	Source Type	Description
[A|M|P]CMAT	I/O API NetCDF	A, M, P	Multiplicative control matrix from **Cntlmat**
[A|M|P]GMAT	I/O API NetCDF	A, M, P	Gridding matrix from **Grdmat**

File Name	Format	Source Type	Description
`[A|M]GSUP`	ASCII	A, M	Supplementary file from **Grdmat** that contains all gridding surrogate codes assigned to each source
`[A|M|P]PMAT`	I/O API NetCDF	A, M, P	Growth matrix from **Cntlmat**
`[A|M|P]RMAT`	I/O API NetCDF	A, M, P	Mass-based or mole-based reactivity matrix from **Cntlmat** (optional, *not yet supported by SMOKE*)
`[A|M|P]SMAT`	I/O API NetCDF	A, M, P	Mass-based or mole-based speciation matrix from **Spcmat**
`[A|M|P]SSUP`	ASCII	A, M, P	Supplementary file from **Spcmat** that contains all speciation profile codes assigned to each source
`[A|M|P]TMP`	I/O API NetCDF	A, M, P	Temporally allocated emissions from **Temporal**
`[A|M|P]TSUP`	ASCII	A, M, P	Supplementary file from **Temporal** that contains all monthly, weekly, and diurnal profile codes assigned to each source. **This file will only work when profiles have not been applied by pollutant.**
`AREA`	SMOKE map	A	Intermediate area-source emissions inventory from **Smkinven**
`COSTCY`	ASCII	A, M, P	Country, state and county file with time zones and, optionally, population
`ELEVTS_[L|S]`	ASCII	P	Elevated point-source file written by **Smkmerge** for REMSAD, UAM, and CAM$_X$ modeling. See `MRG_GRDOUT_UNIT` environment variable setting in Section 7.2.3.2, "Input Environment Variables" [308].
`GEO-CODE_LEVEL[1-4]`	ASCII	A, B, M, P	Geographic level codes and descriptions (also contains Time Zone for GEOCODE_LEVEL4) note: used if USE_EXP_GEO_CODES set to Y and COSTCY file is ignored. See more detail about GEOCODE_LEVELS in Section 8.10.3, "`GEOCODE_LEVEL[1-4]`: `Geographical Input Files`" [447].
`GRIDDESC`	ASCII	A, M, P	Contains description of the modeling grid
`MACTDESC`	ASCII	A, P	MACT code descriptions
`MOBL`	SMOKE map	M	Intermediate mobile-source emissions and/or activity inventory from **Smkinven**
`NAICSDESC`	ASCII	A, P	NAICS code descriptions
`ORISDESC`	ASCII	A, P	ORIS ID code descriptions
`PELV`	ASCII	P	Elevated point-sources identification file; if reporting based on elevated point sources
`PLAY`	I/O API NetCDF	P	Layer fractions file output by the **Laypoint** program
`PNTS`	SMOKE map	P	Intermediate point-source emissions inventory from **Smkinven**
`REPCONFIG`	ASCII	A, M, P	Report configuration file
`SCCDESC`	ASCII	A, M, P	SCC descriptions
`SICDESC`	ASCII	A, M, P	SIC descriptions

7.2.3.2. Input Environment Variables

- `USE_EXP_GEO_CODES`: [default: N]

  If set to Y, this variable is used to specify the use of expanded 12-character geograpic codes contained in the **GEOCODE_LEVEL[1-4]** input files. For **Smkreport**, it controls which ASCII GEOCODE file to use. The user can select the data extraction summary format in the `REPCONFIG` file. If the REPCONFIG file contains `CARB_QADEF` setting, then **Smkreport** creates reports in the CARB QA Data Extraction Format. If the REPCONFIG file contains the `BY GEOCODE_LEVEL1[NAME]` or `BY GEOCODE_LEVEL2[NAME]` or `BY GEOCODE_LEVEL3[NAME]` or `BY GEO-CODE_LEVEL4[NAME]` then **Smkreport** creates reports at each geographic level with its description and local time zone.

- `MRG_GRDOUT_UNIT`: [default: blank]

  This variable is the same one used to control the **Smkmerge** output units. For **Smkreport**, it controls which ASCII elevated file to use, when the ASCIIELEV instruction is included in the `REPCONFIG` file. If the first letter of this setting is an "m" (for mole), then **Smkreport** opens the `ELEVTS_L` file to use in the report; otherwise, **Smkreport** opens and uses the `ELEVTS_S` file. This setting does not do anything to the behavior of **Smkreport** unless the ASCIIELEV setting is included in the `REPCONFIG` file. The default setting of `MRG_GRDOUT_UNIT` results in the `ELEVTS_S` file being used by **Smkreport**.

- `REPORT_ZERO_VALUES`: [default: N]

  Determines if **Smkreport** should keep report entries whose data values are all zero

  - Y: Print entries whose values are zero

  - N: Do not print entries whose values are zero

7.2.3.3. Output Files

File Name	Format	Description
LOGFILE	ASCII	Log generated from executing **Smkreport**
REPORT[#]	ASCII	Multiple ASCII reports. **Smkreport** will output file names of REPORT1, REPORT2, REPORT3, etc., if no other file names are given with the /NEWFILE/ instruction (see the REPCONFIG file documentation in Section 7.3, "REPCONFIG Input File" [308]). If the /NEWFILE/ instructions include physical file names, **Smkreport** will open those files. If the /NEWFILE/ instructions include logical file names (REPORT1, etc.), **Smkreport** will attempt to find the value of the logical file name in the run-time UNIX shell environment. It the logical file name is defined, **Smkreport** will open the physical file associated with the logical file name; otherwise **Smkreport** opens a physical file with the same name as the logical file name.

7.3. REPCONFIG Input File

The `REPCONFIG` file is used to control the **Smkreport** program. As noted earlier, each run of **Smkreport** is constrained to a single source category (area, mobile, or point), at most one speciation scenario, at most

one grid, and at most one hourly emissions file. All of these component datasets are optional, however, and their use is controlled by the settings in REPCONFIG. This file also controls the number and types of reports created by **Smkreport**, which can create an unlimited number of reports in each program run.

The file works on the principles of "packets" and "instructions". Packets are groups of instructions or a single action that have been predefined by the REPCONFIG file format to control the **Smkreport** outputs; instructions are settings that tell **Smkreport** what to do. Some types of packets must appear in one of three sections, while other packets can appear anywhere in the REPCONFIG file. The three sections are: "One-time" (see Section 7.3.1, "One-time section" [309]), "Define-group" (see Section 7.3.2, "Define-group section" [310]), and "Create-report" (see Section 7.3.3, "Create-report Section" [312]); each file section has a large number of options. In this section of the chapter, we first provide details on these three file sections and their packets, and then we describe four additional packets that can appear between other packets at any point in the REPCONFIG file.

The full list of packets that one can use in the REPCONFIG input file are as follows:

One-time section (Section 7.3.1, "One-time section" [309])

- SMK_SOURCE

Define-group section (Section 7.3.2, "Define-group section" [310])

- /DEFINE GROUP REGION/

- /DEFINE SUBGRID/

Create-report section (Section 7.3.3, "Create-report Section" [312])

- /CREATE REPORT/

Packets that can appear anywhere in the file

- /DELIMITER/ (Section 7.3.4, "The /DELIMITER/ packet" [324])

- /AVEDAY/ (Section 7.3.7, "The /AVEDAY/ packet" [325])

- /REPORT TIME/ (Section 7.3.6, "The /REPORT TIME/ packet" [325])

- /NEWFILE/ (Section 7.3.8, "The /NEWFILE/ packet" [325])

- /SET LABEL/ (Section 7.3.5, "The /SET LABEL/ packet" [324])

Comments can be put in the file using a pound sign (#) at the first column of a line, or a double pound sign (##) anywhere else on a line. The double pound can appear after packet labels or instructions in the file.

Finally, note that the maximum number of characters that can be read on any line in this file is 300. This can be important in particular for the SELECT DATA instruction.

7.3.1. One-time section

The first section of the file is the One-time section, and it is a required section. This section is used for setting the environment variables that control **Smkreport** for all reports in a given run. The variables are set by listing the environment variable as an instruction, followed by its value. No **setenv** command (used in the C shell) is needed, because SMOKE is interpreting these values directly. At this time, the one instruction available in the One-time section is the SMK_SOURCE variable:

SMK_SOURCE: Set as "A", "M", or "P", for area-source reporting, mobile-source reporting, and point-source reporting, respectively.

An example of the way this setting would appear in the REPCONFIG file is the following:

```
SMK_SOURCE    A
```

7.3.2. Define-group section

The second section of the REPCONFIG file is the Define-group section. This optional section is used for defining groups of source characteristics for use later in the Create-reports section. The Define-group section uses packets for defining groups. There are currently two types of groupings, and therefore two packet types for this section: regions and subgrids.

7.3.2.1. /DEFINE GROUP REGION/ packet

The /DEFINE GROUP REGION/ packet is used for defining groups by the country, state, and county codes in the inventory. The packet contains the following entries.

- /DEFINE GROUP REGION/ <Label>

  Must appear to start the packet. A label (see below) is required for naming the group.

- INCLUDE and/or EXCLUDE instruction(s) (see below). These are optional.

  Country, state, and county codes in format NSSCCC, where N is the country number, SS is the state code, and CCC is the county code. Zeros can be used in place of CCC to select all counties in a state, or in place of SSCCC to select all counties and states in a country.

- /END/

  Must appear to end the packet.

7.3.2.2. /DEFINE SUBGRID/ packet

The /DEFINE SUBGRID/ packet is used for defining subdomains based on cell numbers from the full grid being applied using the gridding matrices. The packet contains the following entries.

- /DEFINE SUBGRID/ <Label>

  Must appear to start the packet. A label (see below) is required for naming the subgrid.

- INCLUDE and/or EXCLUDE instruction(s) (see below) These are optional.

  (X cell, Y cell):

  Cell numbers should appear for all cells in the subgrid. The count (i.e. the x-cell count and the y-cell count) of the cells depends on the grid defined by the gridding matrix being used by **Smkreport**. A range can be given with the following syntax:

  (X cell, Y cell) TO (X cell, Y cell)

  If only one cell is to be specified, the range must still appear, with both entries indicating the same cell.

Smkreport is not very sensitive to the spacing of this instruction, and additional spaces or no spaces may be placed between the parentheses, the cell numbers, and the commas. Spaces around the "TO" are recommended.

The maximum width of these lines is 100 characters. Multiple entries of this type may appear in the packet, allowing for the grouping of spatially unconnected cells/regions or non-rectangular regions.

- /END/

Must appear to end the packet.

7.3.2.3. Group labels

Group labels, denoted above by <Label>, are used to name a group for identification when creating a report in the Create-report section. The text appearing in the placeholder <Label> can be any alphanumeric combination of characters up to 200 characters, including intervening blanks. The text string (less trailing blanks) that occurs after /DEFINE .../ and before a comment (indicated with ##), is treated as the group label. References to the group label in the Create-report section do not have to be consistent in the case of the text; SMOKE will ignore the case of the text, treating lower- and uppercase characters as if they were the same.

7.3.2.4. INCLUDE and EXCLUDE instructions

The INCLUDE and EXCLUDE instructions are used for defining the meaning of the entries in the packet. If the INCLUDE instruction appears, then **Smkreport** interprets the subsequent entries in the packet as identifying records to *include* in the output report(s). On the other hand, if the EXCLUDE instruction appears, then **Smkreport** interprets the subsequent entries in the packet as identifying records to *exclude* from the output report(s). The INCLUDE and EXCLUDE instructions may appear in both the /DEFINE GROUP REGION/ and /DEFINE SUBGRID/ packets. These instructions act as toggle switches, and **Smkreport** will give a warning if a county previously included is later excluded. All packet entries after an instruction will be interpreted according to the instruction, until another instruction is given, or until the packet ends. The INCLUDE instruction is the default, so if neither instruction is given, SMOKE will assume that the records in the packet are those to be only in the report.

The ordering of INCLUDE or EXCLUDE instructions in a group definition can change the meaning of the group definition. If INCLUDE is given first, **Smkreport** will initialize the group to exclude all grid cells and then proceed with interpreting the instructions. On the other hand, if EXCLUDE is given first, **Smkreport** will initialize the group to include all grid cells. Here is an example for a small 6-cell by 6-cell grid, which shows how the order of the INCLUDE and EXCLUDE statements changes the meaning of a group definition.

If the instructions are

```
/DEFINE SUBGRID/ Example subgrid
    INCLUDE
      (2,2) TO (6,6)
    EXCLUDE
      (3,3) TO (5,5)
    INCLUDE
      (4,4) TO (4,4)
/END/
```

then the following cells (shown in gray) will be selected:
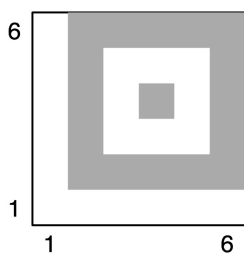
If the instructions were rearranged to have the EXCLUDE statement first, then the initialized status for all cells would be INCLUDE. The additional INCLUDE statements are therefore unneeded, and the definition of the group changes to the following:
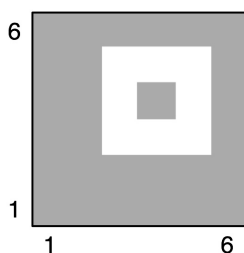
7.3.2.5. Future grouping options

As noted above, the REPCONFIG file currently can have only two possible groups in the Define-group section (regions and subgrids) In the future, other types of groups could be permitted if a need for them is identified:

- /DEFINE GROUP SCC/: Permits user-defined SCC groups

- /DEFINE GROUP PLANT/ Permits user-defined groups by country/state/county code and plant ID

- /DEFINE GROUP STACK/ Permits user-define groups by country/state/county code, plant ID, and stack ID

- /DEFINE GROUP POINT/ / Permits user-define groups by country/state/county code, plant ID, stack ID, and point ID

- /DEFINE GROUP SOURCE/: Permits user-defined source groups by SMOKE source ID

- /DEFINE GROUP HOURS/: Permits user-defined hour groupings

7.3.3. Create-report Section

The third section of the REPCONFIG file is called the Create-report section. It is responsible for providing instructions to **Smkreport** for generating the reports. The /CREATE REPORT/ packet is used in this section. This packet uses the following types of instructions (listed in alphabetical order):

- ARRANGE

- ASCIIELEV

- BY

- CONTROL

- GRIDDING

- LAYFRAC

- NORMALIZE

- NUMBER

- PROJECTION

- SELECT

- SPECIATION

- TEMPORAL

- TITLE

- UNITS

By using these instructions (some of which have a number of different options), many different types of outputs can be achieved by **Smkreport**. The ordering of the instructions in the packet is not important, *with one exception*: if multiple titles are used, they will be printed to the report in the order in which they appear in the packet. The remainder of this subsection provides descriptions of each of these instructions.

7.3.3.1. ARRANGE instruction

The ARRANGE instruction controls the arrangement of variables and files for a single report. This instruction sets how many variables are included in a report or a section of a report, and can set the number of sections that a report will have. It also permits a database format that includes an emissions data value for each record in the report.

If no ARRANGE instruction is provided for a report, **Smkreport** will attempt to write all of the variables requested in the report to the same file and section. If too many variables are requested in the report, this could cause **Smkreport** to abort with an error. *In general, if you have more than 20 variables that will be written to a given report, you should consider using the ARRANGE instruction.*

The syntax of the ARRANGE instruction is:

ARRANGE <option>

with the following valid options:

- MULTIFILE [*number of variables*]: Writes the report into multiple output files. This is different from the **Smkreport** capability of creating multiple reports: the instruction applies to a single report and indicates that the report should be output as multiple files. The file names in this case are created automatically by **Smkreport** based on the file name set by the associated NEWFILE instruction (explained in Section 7.3.8, "The /NEWFILE/ packet" [325]), by appending a "_1", "_2", "_3", and so forth to the file name that would otherwise be output by **Smkreport**. For example, if the output report setting REPORT1 is defined as /outdir/test/report1.myfile.txt, but that report requires two files, then the actual files that would be written would be /outdir/test/report1.my-file.txt_1 and /outdir/test/report1.myfile.txt_2.

The *number of variables* option controls the number of variables written to each file of the report. For example, if the report will include 100 variables and the number of variables is set to 15, then **Smkreport** will create a report using seven separate files, six that include 15 variables (taking care of the first 90 variables) and a seventh file that has the remaining 10 variables. *Smkreport allows a maximum of 30 variables per file using the MULTIFILE instruction.*

- ONEFILE [*number of variables*]: Writes the report into a single output file with multiple sections. Like the MULTIFILE option, this is different from the **Smkreport** capability of creating multiple reports: the instruction applies to a single report, but permits multiple sections of the report to be written to the same file.

  The *number of variables* option controls the number of variables written to each section of the report. The number of sections is set automatically by dividing the number of data variables selected for the report by the number of variables.

- DATABASE: Writes the report one variable at a time, with only one data variable on each row of the report. The attributes of the row are repeated for all variables, and all entries are written for one variable before the next variable is written. For example, if a report BY SCC10 is requested for the variables NOX and SO2, the report will list all SCCs with the NOX emissions as a data variable, followed by all SCCs again with SO2 emissions. Without the DATABASE instruction, the NOX and SO2 values would be written to the same line of the report and the SCC codes would be written only once.

7.3.3.2. ASCIIELEV instruction

The ASCIIELEV instruction causes **Smkreport** to use the elevated ASCII emissions file (`ELEVTS_L` or `ELEVTS_S`) written by **Smkmerge** as the sole source for emissions information in the report. This option can be used only for point sources (SMK_SOURCE is set to P). When this option is included in a `REPCONFIG` file, all other reports in the `REPCONFIG` file must also include the ASCIIELEV instruction, or **Smkreport** will produce an error.

Using this instruction, **Smkreport** will use the emissions in the `ELEVTS_S` or `ELEVTS_L` file as the sole source of emissions and inventory information. These files include information about the states, counties, plants, stacks, stack parameters, hour, cell number, and speciated emissions. Therefore, only certain instructions may be used with the ASCIIELEV instruction, or else a warning will be printed to the log file and the unusable instruction will be ignored. The following subset of /CREATE REPORT/ instructions can be used with the ASCIIELEV instruction:

- ARRANGE

- BY <COUNTRY | STATE | COUNTY | CELL | ELEVSTAT | HOUR | PLANT | SOURCE | STACK>; other BY options will be ignored

- NORMALIZE

- NUMBER

- SELECT

- TITLE

- UNITS

The following instructions are implied by the ASCIIELEV instruction:

- CONTROL (implied only if the emissions in the elevated ASCII file are controlled)

- GRIDDING

- PROJECTION (implied only if the emissions in the elevated ASCII file are grown)

- SPECIATION [MASS|MOLE] (which option is used depends on the `MRG_GRDOUT_UNIT` setting, described below)

- TEMPORAL

7.3.3.3. BY instruction

The BY instruction controls what columns in addition to data columns will be printed in the output report. Multiple BY commands may appear in the same /CREATE REPORT/ packet, but not all combinations make sense. The BY instructions tell the program the basis on which to sum the emissions or activity values. The syntax of the BY instruction is:

BY <option>

with the following valid options:

- CELL: Records are printed by cell. This option requires the GRIDDING instruction, so **Smkreport** automatically adds the GRIDDING instruction when the BY CELL instruction is used (even if it is not actually in the `REPCONFIG` file).

- COUNTRY [NAME]: Records are printed by country total. This option invokes use of the `COSTCY` file. NAME can be appended to the option to additionally output the country name instead of only the country number.

- COUNTY [NAME]: Records are printed by county total. This option invokes use of the `COSTCY` file. NAME can be appended to the option to additionally output the county name instead of only the county number.

- DOMCODE: Records are printed by Day of Month profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- ELEVSTAT [STACKGROUP]: Records include the status of sources as elevated (E), plume-in-grid (P), and low-level (L). All elevated statuses could appear in the same report, depending on the inventory and elevated source selection criteria. `STACKGROUP` can be appended to the option to additionally output stack group number from `PELV`. This instruction can be used for point sources only.

- FRICODE: Records are printed by Friday diurnal profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- GEOCODE_LEVEL1: Records are printed by the Country-level geographic code.

- GEOCODE_LEVEL2: Records are printed by the State-level geographic code.

- GEOCODE_LEVEL3: Records are printed by the County-level geographic code.

- GEOCODE_LEVEL4: Records are printed by the Tribal-level geographic code.

- HOUR: Records are printed by hour total for each day. This option implies the TEMPORAL instruction, so **Smkreport** automatically adds that instruction even if it was not included in the input.

- INTEGRATE: Records are printed by the integration status of sources. "Y" willl appear for the integrated sources that were used to calculate the NONHAP[VOC|TOG] inventory pollutant.

- LAYER: For point sources, daily emissions are printed by vertical layer. This option implies the LAYFRAC option (discussed later) that requires input of the PLAY SMOKE intermediate file. In the report, records are written first for one layer, then the next, and so on. Only records with nonzero emissions in a layer are written. When used to together with the HOUR instruction (above), emissions will be written by hour as well as by layer. Since the number of records the report when using HOUR instruction is very large, we recommend that a source group be used to subselect the inventory records when appropriate.

- MACT [NAME]: Records are printed by MACT code. NAME can be appended to the option to additionally output the MACT code description (from the MACTDESC file) instead of only the MACT code.

- MATBURNED: For wildfire point sources, records are printed by MATBURNED code.

- METCODE: Records are printed by Meteorology-based hourly profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- MONCODE: Records are printed by Mothly profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- MNDCODE: Records are printed by Monday diurnal profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- NFDRSCODE: For wildfire point sources, records are printed by NFDRSCODE code.

- NAICS [NAME]: Records are printed by NAICS code. NAME can be appended to the option to additional output the NAICS code description (from the NAICSDESC file) instead of only the NAICS code.

- ORIS [NAME]: Records are printed by ORIS ID code. NAME can be appended to the option to additional output the ORIS ID code description (from the ORISDESC file) instead of only the ORIS ID code.

- PLANT [NAME]: For point sources, emissions are printed by plant total. This option cannot be used in conjunction with the BY SOURCE option; if the REPCONFIG file contains both instructions, SMOKE will ignore the BY PLANT instruction.

  When the NAME option is included, the plant name will be included in the report. Otherwise, only the plant number will be included.

- ROADCLASS: Records are printed by road class total. This option can only be used for mobile sources.

- SATCODE: Records are printed by Saturday diurnal profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- SCC [N] [NAME]: Records are printed and summed by all or part of the SCC, depending on the value of N, which can be set to 1, 2, 3, or 4. These numbers correspond to the SCC levels described in Section 2.3.5, "Source Classification Codes" [14]. When N = 4, the full SCC is used. When N = 1, 2, or 3, the records are summed and reported by the corresponding SCC level. Differences between point and nonpoint/mobile SCCs are handled as described by the SCC structures. If no value is given. N is assumed to be 4 by default.

  The NAME option can be used for any N value. The SCC description is truncated assuming that semicolons separate the different parts of the SCC descriptions. If there are more than 3 semicolons in an SCC description (dividing into 4 parts), then all of the SCC sections after part 3 are put into part 4. The SCC description written by **Smkreport** will match the value of N by including all parts of the SCC description up to and including the value of N.

- SCC10 [NAME]: This option maintains backward compatibility with previous versions of **Smkreport**. It is the same as the option "SCC 4 [NAME]".

- SIC [NAME]: Records are printed by SIC totals.

  When the "NAME" option is included, the SIC name will be included in the report. Otherwise, only the SIC code will be included. As with the BY SCC10 NAME instruction, the BY SIC NAME instruction will change the column delimiter from a semi-colon to the pipe symbol ("|"). The SIC name will also be enclosed in double quotes.

- SOURCE [STACKPARM] [NAME] [LATLON]: Records are printed by source totals. Source characteristics are printed as well:

  - Point: country/state/county code, plant ID, stack ID, point ID, segment ID, SCC, and SIC

  - Area: country/state/county code and SCC

  - Mobile: country/state/county code, road type code, vehicle type code, link ID, and SCC

  For area and mobile sources, the SCC descriptions will automatically be included in the report. This is the same as adding the "BY SCC10 NAME" instruction to the report instructions. There is no way to turn off this feature.

  For point sources, STACKPARM can be appended to the option to also output the stack parameters (height, diameter, exit velocity, exit temperature). Also for point sources, NAME can be appended to the option to get the facility names in the output file. LATLON can be appended to the option to print each source's x- and y-coordinates.

- SPCCODE <POLLUTANT>: Records are printed by speciation profile codes assigned to sources by the speciation cross-reference file. *Only one pollutant can appear with this command throughout the entire* `REPCONFIG` *file.* The pollutant should be one that is in the inventory file and that was processed as part of the chemical speciation file run. This BY option does not imply the SPECIATION instruction (see below), and it can be used without included speciated emissions in the report. For example, the BY SPCCODE instruction may be used to indicate which speciation profiles were assigned to total VOC emissions (without reporting speciated VOC emissions). **NOTE: this BY SPCCODE option can't be used more than once in a REPCONFIG file.**

- SRCTYPE: Records are printed by source type code.

- SRGCODE [FALLBACK]: Records are printed by surrogate codes assigned to sources by the gridding cross-reference file. If no option is used, both primary and fallback surrogate codes are included in the output report. If the FALLBACK option is used, only the fallback surrogate codes are included in the output report, and a zero is used for all records that did not use a fallback surrogate (i.e., their primary surrogate did not cause emissions to go to zero). This BY option does not imply the GRIDDING instruction (see below); therefore, it can be used without applying the gridding matrix in the report. A value of -9 is used for fallback surrogate IDs in the resulting report.

- STATE [NAME]: Records are printed by state totals. This option invokes the use of the `COSTCY` file. NAME can be appended to the option to additionally output the state name instead of only the state number.

- SUNCODE: Records are printed by Sunday diurnal profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- THUCODE: Records are printed by Thursday diurnal profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- TUECODE: Records are printed by Tuesday diurnal profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

- WEKCODE: Records are printed by Weekly profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant. Note that this option spelling is *not* WEEKCODE.

- WEDCODE: Records are printed by Wednesday diurnal profile code. This option will work correctly only when monthly temporal profiles have not been applied by pollutant.

The most detailed BY instruction will determine the resolution of the emissions and activity totals. For example, the COUNTRY, STATE, and COUNTY options can be used in the same report to affect the column headers, but the data output will not be different because the emissions totals will be at the finest resolution specified. In other words, if the instructs are to report BY COUNTY and BY STATE, the emission totals in that report will be by county, and the BY STATE instruction will merely add the state code to the columns printed. Another example is that the BY SCC10 instruction will override the resolution of BY ROADCLASS.

Future versions of the BY command may support the following options, if a need for them is identified by SMOKE users:

- STACK [NAME]: For point sources only, records are printed as facility and stack totals. NAME can be appended to the option to get the facility names in the output file.

- POINT [NAME]: For point sources only, records are printed as facility, stack, and point totals. NAME can be appended to the option to get the facility names in the output file.

- VTYPE: For mobile sources only, records are printed as vehicle-type totals.

- GROUP <option>: Records are printed based on the user-defined groups, with the following options:

  - REGION: Rows in report are user-defined region groups.

  - SCC: Rows in report are user-defined SCC groups.

  - SOURCE: Rows in report are user-defined source groups, by SMOKE record number.

  - HOURS: Rows in report are user-defined hour groups.

7.3.3.4. CONTROL instruction

The CONTROL instruction applies the multiplicative control matrix to all pollutants and species requested in the report. This instruction does not limit the options available for resolution of the report, so reports that have the CONTROL instruction can also include any combination of BY instructions. **Smkreport** is unable to apply the reactivity control matrix at this time.

7.3.3.5. GRIDDING instruction

The GRIDDING instruction controls whether or not the gridding matrix is used by **Smkreport**. When the instruction is present, the gridding matrix is used in generating the report; the matrix is applied to the emissions before the totals are made. The gridding matrix file name is determined by the environment variables set at run time. One use of the GRIDDING instruction would be the following: a report could be generated without GRIDDING to see emission totals before gridding, and another report could be generated with GRIDDING to see the totals after gridding. The totals from the two reports could be compared to see if any emissions were lost because of the gridding step.

The GRIDDING instruction is assumed when any of the following other instructions appear in the packet:

- SELECT SUBGRID

- BY CELL

- NORMALIZE CELLAREA

7.3.3.6. LAYFRAC instruction

The LAYFRAC instruction controls whether or not the point source layer fractions file (PLAY) from the **Laypoint** program is used in determining elevated and low-level sources. If the PLAY file is not used for determining elevated sources, the PELV file from the **Elevpoint** program will be required.

7.3.3.7. NORMALIZE instruction

The NORMALIZE instruction controls whether or not the emission values will be normalized using some other data value. Two normalize options are currently supported; CELLAREA and POPULATION. Emissions normalization can provide additional insight to the inventory, more than can be obtained by simply summarizing emissions values.

The syntax of the NORMALIZE instruction is:

NORMALIZE <option>

with the following valid options:

- CELLAREA: Normalize emissions based on grid cell area. When the emissions are normalized by cell area, they are divided by the area of each cell (or just the cells selected, if a subgrid is used). Normalizing by grid cell area is useful for comparing emission reports from grids at different resolutions, including nested grids, to ensure that a consistent emissions density is maintained across the multiple grid resolutions.

  For Lambert and UTM grids, the cell ares are computed using the formula:

  $DX * DY$

  where

  - DX is the cell x-direction size in meters

  - DY is the cell y-direction size in meters

  For lat-lon grids, the area of the lat-lon cell are estimated using the formula:

  $(\cos Y)*(\pi * DX * R / 180)*(\pi * DY * R / 180)$

  where

  - DX is the cell x-direction size in degrees longitude

  - DY is the cell y-direction size in degrees latitude

  - Y is the cell center in degrees latitude

  - R is the radius of the earth in meters

The NORMALIZE CELLAREA command cannot be used with polar stereographic grids at this time.

This instruction requires that the GRIDDING instruction also be used, because the gridding information is needed in order to normalize the emissions by grid area. If the GRIDDING instruction is left out of the report instructions, **Smkreport** will add it to the report and continue.

- POPULATION: Normalize emission based on county population. Emissions in the report are divided by county population, regardless of whether the BY COUNTY instruction has been used in the report instructions.

  This option requires that the COSTCY input file to **Smkreport** include an additional field and a #POPULATION header (as is described in the COSTCY documentation in Chapter 8, *SMOKE Input Files* [345]).

Future versions of the NORMALIZE instruction may support the following option, if a need for this option is identified by SMOKE users:

- NORMALIZE COUNTYAREA: Normalize emissions based on county area.

7.3.3.8. NUMBER instruction

The NUMBER instruction controls the format used to display *data values* in the report. The argument for the number instruction applies to all data values that are written to the report, and the structure of the argument is simply the formats supported by Fortran. The data values in the reports are rounded to the nearest value allowed based on the format selected. A value of 5 is rounded up to the next decimal place, when such rounded is needed.

The syntax of the NUMBER instruction is simply:

NUMBER <format>

where <format> is the Fortran format specification. The common formats you might use are the following:

- Float format: F<x>.<y>: where <x> is the number of digits for the number, including the decimal place, and <y> is the number of digits shown after the decimal place.

- Exponential format: E<x>.<y>: where <x> is the complete width of the value, including five characters for "E+00" and the decimal, and <y> is the number of significant figures.

If the NUMBER instruction does not appear in the report instructions, the E8.3 format is used.

7.3.3.9. PROJECTION instruction

The PROJECTION instruction will apply the growth matrix to all pollutants and species requested in the report. This instruction does not limit the options available for resolution of the report, so reports that have the PROJECTION instruction can also include any combination of BY instructions.

7.3.3.10. SELECT instruction

The SELECT instruction controls which records in the inventory are used in generating the report. It allows the user to extract particular records from the inventory, for inclusion in the report. This instruction also permits references to the groups defined in the Define-groups section of the REPCONFIG file. If no SELECT instruction is used, then the other report instructions will be applied to all records in the inventory. Only one SELECT statement of each type is permitted in per report; for example, only one SELECT REGION statement and one SELECT SUBGRID statement could be used for a single report. Of course, different

SELECT REGION or SELECT SUBGRID instructions could be used for different reports in the same output file.

The syntax of the command is:

SELECT <options> <label>

where <label> is an optional field, depending on the <option> selected. The following options are available for the SELECT instruction:

- DATA <label>: Selects data values to include in the report. When no SELECT DATA instruction is used, all pollutants, activities, or species found in the input file(s) are reported. When species are being reported, the corresponding pollutants in the inventory or hourly emissions file are reported as well as the speciated emission values. The <label> must be a space-delimited list of pollutant, activity, or species names. Do not include the pollutant names in the list when species are listed, as these will be included by default.

  To specify a species value when the species name is the same as the pollutant name, an "S" and a hyphen can be inserted before the species name. You can also use this method for specifying the sum of post-speciated emissions. For instance, to specify carbon monoxide after speciation, the SELECT instruction should be:

  SELECT DATA S-CO

  To specify the sum of VOC emissions after speciation, the SELECT instruction should be:

  SELECT DATA S-VOC

  *Note that **Smkreport** does not determine whether the summation makes sense, and does not correct for species molecular weights.* You must ensure that the data you request makes sense.

  Also note that the maximum number of characters that can be read on any line in this file is 1024. This can limit the number of pollutants that you can specifically request in a SELECT DATA statement.

- ELEVATED [PING]: Selects all elevated point sources (recognized for point sources only). If no elevated sources are found (after applying all other selection criteria), the report will contain a warning to that effect. If the PING option appears, only plume-in-grid sources will be selected. When the ELEVATED instruction is used by any report, the PELV file (from **Elevpoint**) or PLAY file (from **Laypoint**) will be required to determine which sources are elevated. The latter will be used when the LAYFRAC instruction is also included in the report instructions. If the PING option is used, the PELV file will be required in all cases, although the LAYFRAC instruction can still be used to set the elevated source status based on the layer fractions.

- NOELEVATED: Selects all low-level point sources (recognized for point sources only). If both the ELEVATED and NOELEVATED commands appear in the same report, a warning will be given, and the latter setting will be used to generate the report. If no low-level sources are found (after applying all other selection criteria), the report will contain a warning to this effect. The same rules apply for use of the PELV or PLAY files as described for the ELEVATED option.

- REGION <label>: Selects a region using the country/state/county code or a region group. For this option, <label> must be either a country/state/county code in NSSCCC format, or a region group label from the Define-group section of the REPCONFIG file.

- SUBGRID <label>: Selects a subgrid using the grid cell numbers or a subgrid group label. For this option, <label> must be either a range of grid cells, or a subgrid label from the Define-group Section of the file. When the <label> defines the cell range, you should use the same syntax that was specified

above for the /DEFINE SUBGRID/ packet. However, the INCLUDE setting is assumed, and **Smkreport** will not understand the INCLUDE and EXCLUDE commands as part of the <label>.

For the REGION and SUBGRID options, the value of <label> must either match a group or subgrid label or match characteristics of the inventory or grid. Also, the <label> for the REGION and SUBGRID options must be 200 characters or fewer.

Future versions of **Smkreport** may support the following additional options, if a need is identified:

- SCC <label>: Select a single SCC or SCC group for reporting.

- PLANT <label>: Select a single point source plant, or a plant group for reporting.

- STACK <label>: Select a single point source stack, or a stack group for reporting.

- POINT <label>: Select a single point source point, or a point group for reporting.

- HOURS <label>: Select a single hour or hour group for reporting.

- SOURCE <label>: Select a single source or source group for reporting.

7.3.3.11. SPECIATION instruction

The SPECIATION instruction controls whether the speciation matrix is used by the program. This option controls the columns of data that will be printed in the output file. When the SPECIATION instruction is in the report instructions, the columns in the output file will include speciated emissions values as well as inventory emissions from the SMOKE inventory or hourly emissions file (depending on whether or not the TEMPORAL instruction is used). When the SPECIATION instruction is not used, the columns in the output file will be inventory emissions and/or activities.

The syntax of the SPECIATION instruction is:

SPECIATION <option>

where the following options are supported

- MOLE: Uses the mole-based speciation matrix.

- MASS: Uses the mass-based speciation matrix.

7.3.3.12. TEMPORAL instruction

The TEMPORAL instruction controls whether or not emission values are used from the hourly emission files ATMP, MTMP, or PTMP. When this instruction is used, the default behavior of **Smkreport** is to sum up the emissions by day and report emissions as daily totals. To override this behavior, the BY HOUR instruction must be used.

When dates and times are given in the report, the hours are reported in the time zone from the ATMP, MTMP, or PTMP file. Hours are reported as *standard* time values for the time zone stored in the header of the hourly emissions file; hours are never reported as *daylight* time values.

The TEMPORAL instruction is assumed when either of the following options appear in the report instructions:

- SELECT HOURS

- BY HOUR

In the future, we may be able to enhance this instruction to permit the reported hour values to be labeled with a different time zone than is in the hourly emissions file.

7.3.3.13. TITLE instruction

The TITLE instruction allows one or more titles to be added to a report. The titles are printed into the report in the order in which they appear in the packet. Quotes are *not* needed in titles with blank spaces. The title will be ended at the last character on the line, or at a double pound sign (##), whichever appears first on the line. Double pound signs cannot be printed in the titles of reports. It is recommended that titles be 80 characters wide. The maximum title width is 300 characters. The syntax of the TITLE instruction is the following:

TITLE <title text>

7.3.3.14. UNITS instruction

The UNITS instruction allows you to control the units of the output data. The current version of **Smkreport** does not permit you to vary the units by data value; instead, the same units must be used for all data values, or the default units will be used. Future versions of **Smkreport** may permit control of the units by data variable.

All unit conversions that *can* be done, will be done. For example, if the inventory is stored in tons/hour, and the output is requested in fathoms/day, the hour-to-day conversion will be made, while the impossible tons-to-fathoms conversion will not be made. The syntax of the UNITS instruction is the following:

UNITS ALL <units>

Valid values for the <units> option numerator are the following:

- ton(s)

- g, gm(s), gram(s)

- kg(s)

- mole(s), gm mole(s)

Valid values for the <units> option denominator are the following:

- year(s), yr(s)

- day

- hour(s), hr(s)

- min(s)

- day(s)

- s, sec(s)

No quotes are needed around the unit labels, and they must be 30 characters or fewer. Some examples of the units instruction are the following:

```
UNITS ALL gm moles/s
UNITS ALL tons/hr
UNITS ALL kg/year
```

7.3.4. The /DELIMITER/ packet

The /DELIMITER/ packet can be used to change the column delimiter used in the output reports from **Smkreport**. The default delimiter is a semicolon (;). You can change the delimiter to any single-character value to suit your needs, although using a space as a delimiter may result in slightly misformatted reports. Some reasons to change the delimiter from the default are:

- You are reporting "BY SOURCE" and including the source name, and it has semicolons included in one or more of the names. This would prevent those records in the output report from being read into another program using semicolon as the delimeter.

- Software you want to use to read the **Smkreport** output report does not support semi-colons for delimiters.

- You prefer a different appearance in the reports; you can select another delimiter that does not conflict with other fields in your report.

The syntax of this instruction is the following:

/DELIMITER/ ,

This example results in a comma (,) being used as the delimiter for columns of any results from /CREATE REPORT/ instructions that followed this instruction in the REPCONFIG file.

7.3.5. The /SET LABEL/ packet

The /SET LABEL/ packet can be used to set a label to include on every row of every subsequent report defined in the REPCONFIG file. The label text can be either the text that appears in the packet, or the label can be defined by an environment variable. This feature facilitates combining multiple reports from separate SMOKE runs into a single report more easily. For example, if the packet is used to create state reports for all sectors run through SMOKE, then those state reports could be concatenated, the file headers removed (except for the first header), and the individual sectors would be idenfified by the labels on the rows.

The syntax for the packet is as follows:

/SET LABEL/ <label>

where <label> is limited to 200 characters and is one of the following:

- a string of text that will appear near the beginning of every data line of the report. The only invalid string is the word "OFF" in capital letters.

  Example: /SET LABEL/ NonEGU Point

  In this example, all subsequent reports will contain a new column called "Label" and every value in the entire report will be "NonEGU Point".

- an environment variable designated by <label> starting with a dollar sign.

  Example: /SET LABEL/ $RUN_ID

In this example, **Smkreport** will attempt to evaluate the environment variable RUN_ID from the shell running **Smkreport**. If RUN_ID is a defined environment variable (e.g., the shell had set RUN_ID to "oarea"), then all subsequent reports will contain a new column called "Label" and every value in the entire report will be "oarea".

If the environment variable is not defined, set to blank, or otherwise invalid, the instruction will behave like the /SET LABEL/ OFF case below.

• the word "OFF".

Example: /SET LABEL/ OFF

This syntax turns off use of the label in all subsequent reports until a new valid /SET LABEL/ packet is found.

7.3.6. The /REPORT TIME/ packet

The /REPORT TIME/ packet can be used to change the hour value for which daily-total hourly emissions are output. The syntax of this instruction is the following:

/REPORT TIME/ <HHMMSS>

The setting uses HHMMSS format, in which HH is the hour and MMSS is 0000. The time specified is in the time zone of the input hourly data file. This setting is ignored when hourly data are not input to **Smkreport**. The default behavior of **Smkreport** is to report at the last hour of the day, hour 23 (hours are counted from 0 to 23). This setting must be consistent with the SMOKE intermediate files, which use hour numbers 0 through 23 because they are based on the I/O API. However, these hour numbers are different from those included in the report because of the "BY HOUR" instruction, which numbers the hours 1 through 24.

7.3.7. The /AVEDAY/ packet

The /AVEDAY/ packet can be used to toggle on or off the use of the average-day emissions from the SMOKE inventory file. The syntax of the command is the following:

/AVEDAY/ <option>

The <option> field can be set to Y or N. The default behavior of the program (if no /AVEDAY/ packet appears) is to *not* use the average-day emissions in the reports. This packet only affects the program *only* when the inventory emission are being reported. If the TEMPORAL instruction is given, the output report will contain whatever emission values were used to create the hourly emissions file.

7.3.8. The /NEWFILE/ packet

The /NEWFILE/ packet can be used before or between any /CREATE REPORT/ packets to cause **Smkreport** to write out the subsequent reports to a different report file. The syntax of the command is the following:

/NEWFILE/ <Filename>

The placeholder <Filename> is replaced by either:

1. A physical file name, including the path.

2. A logical file name.

Smkreport will detect whether the file is a physical or logical file name by checking to see if the <Filename> value is defined with a nonblank value in the run-time environment. If it is, then **Smkreport** assumes the value is a logical file name, and opens a file using the value found from the environment.

If no information is given for <Filename>, the output files will be named by their automatic logical file names, `REPORT[#]`, where [#] is replaced by the number of the report file in order of its associated /NEWFILE/ packet.

7.4. Reports Created by Smkreport

As described in Section 7.2, "**Smkreport** Program" [305], the **Smkreport** program creates user-defined reports and summaries for QA purposes based on the SMOKE intermediate files (e.g., SMOKE inventory files, hourly emissions, speciation matrix, gridding matrix). The reports that are created by **Smkreport** can easily be imported into a spreadsheet program or into a database because the ASCII output format is delimited. Note that report files larger than about 60,000 lines will not import into Excel because of limitations of that software.

Because **Smkreport** is very flexible, there are a large number of reports that you can generate. The "File Format(s)" section below describes the guidelines that **Smkreport** uses in building the output reports, based on the instructions the user provides to **Smkreport**. This section is followed by a large number of example reports, to help you understand the possible format options.

7.4.1. File format(s)

7.4.1.1. Multiple reports in one file

Each file output by **Smkreport** can contain multiple reports. The user has control over how many reports are included in each file (please see the REPCONFIG file documentation above for more information on using the /CREATE REPORT/ and /NEWFILE/ packets). The reports in a file are separated by a row of pound signs (#). If having multiple reports in a single file causes problems for importing the data into other software, then you should create a REPCONFIG file that specifies a new file for each report.

7.4.1.2. Basic report structure

The reports in an output file use columns and rows to structure the information. The reports are structured to permit the user to import a report into spreadsheet or database software, or some other data-processing tool for further analysis. The columns in the reports are semicolon delimited (though the delimiter can be set by the user), have labels, and have units. Where possible, we have also tried to make sure that the text versions of the reports can be viewed easily. The width of the widest entry in a column is used to set the width for the column, thereby making fixed-file format transfer to other software possible.

7.4.1.3. Advanced report structures

Using the ARRANGE instruction in the **Smkreport** REPCONFIG input file, you can also create reports with more advanced structures: a single report that is composed of multiple physical files; a report that is single physical file but has multiple sections; or a report that uses a "database" format of one data value (e.g., emissions value) per line. Use of these formats is described in the REPCONFIG file documentation in Section 7.3, "REPCONFIG Input File" [308].

7.4.1.4. Structure template

The following is a general template for the **Smkreport** output files. It consists of a REPORTS section and a METADATA section. These two sections are discussed in detail below the template.

```
<REPORT1>
     <User Title(s), one per line>
     <Automatic Titles, one per line>
     <Column headers>
     <Units, for columns that need them>
     <Line of minus signs>
     <Data values for all rows>
#############################################################################
<REPORT2, if any>
     <User Title(s), one per line>
     <Automatic Titles, one per line>
     <Column headers>
     <Units, for columns that need them>
     <Line of minus signs>
     <Data values for all rows>
#############################################################################
METADATA
     <Input files>
     <Echoed group definitions from REPCONFIG file>
     <Echoed report instructions from REPCONFIG file>
<End of file>
```

7.4.1.5. Elements of the REPORTS section

7.4.1.5.1. User titles

The first section of each report is a title section. The first title entries in the report are any titles specified in the /CREATE REPORT/ packets in the REPCONFIG file, created by the user. They are listed in the same order as those provided in the input packets for the report. It is recommended that user titles be no more than 80 characters wide (to more easily fit on printed documents), although they can have up to 300 characters.

7.4.1.5.2. Automatic titles

Automatic titles are inserted by **Smkreport** to indicate the operations that were done by **Smkreport** to generate the requested report. This helps the report be self-describing. The automatic titles are formatted to be no larger than 80 characters wide, in order to permit easier viewing as text files. The following automatic titles are applied, where applicable:

- The source category processed

- The year of the base inventory

- The year of the grown inventory (only if a grown inventory file is input to **Smkreport**)

- Whether growth factors were applied, and if so, for what year

- Whether multiplicative control factors were applied, and if so, for what year

- Whether a gridding matrix was applied, and if so, the grid name

- Whether a speciation matrix was applied, and if so, whether it was mole-based or mass-based

- Whether annual, average-day, or hourly data from SMOKE intermediate files are the basis of the emissions in the report

- If hourly data were input, the time period processed with dates, hours, and time zones

- Whether annual or average-day data were the basis for the report

- Whether the emissions are divided by grid cell area (NORMALIZE CELLAREA instruction)

- Whether the emissions are divided by population (NORMALIZE POPULATION instruction)

- The name of the group used to select the data, if any (the group definition will be listed in the METADATA section)

- The name of the subgrid used to select the data, if any (the subgrid definition will be listed in the METADATA section)

7.4.1.5.3. Data columns written

The particular columns of data that are written to a report are determined by the BY instructions in the /CREATE REPORT/ packet. Typically, one column is created for each BY instruction, which corresponds to the name of the BY instruction. The exceptions to this rule are the following:

- The BY COUNTRY, BY STATE, and BY COUNTY instructions, which output the country, state, and county codes, and optionally the country, state, and county names, respectively (if NAME is included at the end of the instruction).

- The BY SOURCE instruction, which outputs columns for all source characteristics (point sources: country/state/county, plant, char1, char2, char3, char4, char5, SIC; area sources: country/state/county, SCC; mobile sources: country/state/county, road class, vehicle type, SCC). For point sources, the BY SOURCE instruction can also output four additional columns for stack parameters, and another additional column for facility name.

In addition to the columns specified by the BY instructions, the emissions and/or activity data are written out to the report. If no speciation option is used (i.e., the SPECIATION instruction is not given), then the other data columns are the emissions and activities from the SMOKE inventory file (e.g., NOX, VOC, PM10, VMT). If the speciation option *is* used, then the other data columns are the speciated emissions data (e.g., NO, NO2, ALD, PAR, ISOP). The SELECT DATA instruction determines which inventory data and speciated emissions data are written to the reports. When speciated data are output, the columns for the corresponding inventory pollutants are written as well (as explained more fully in the input file description of the SELECT DATA instruction, earlier).

The order of the columns of data values is taken from the order of the data in the SMOKE intermediate files. The orders of pollutants and activities are controlled by the INVTABLE file when **Smkinven** is run to create the SMOKE inventory files. Species are arranged by **Spcmat** based on the order of the pollutants, and then alphabetically for species coming for the same pollutant, and this order is used to create the speciation matrix.

7.4.1.5.4. Column delimiters

The default column delimiter in most cases is a semicolon (;). However, if you select the "BY SCC10 NAME" or "BY SIC NAME" instruction, the SCC or SIC description will be included in the report and

the default delimiter will be a pipe symbol (|). This is because the SCC and SIC descriptions can include semicolons. In either case, you can use the /DELIMITER/ packet to manually set your choice of delimiter, if desired.

7.4.1.5.5. Column headers

Each column of data is labeled with a header name. Where applicable, the columns also are labeled with units (discussed below), as set by the UNITS instruction or by the default units for the data used by the program. For the emissions and activity data, the header names are simply the names of the pollutants or activity variables input to SMOKE. For columns generated by the BY instructions, the following possible column headers are used by SMOKE. These are listed in the order in which they would appear.

- Date (values of MM/DD/YYYY)

- Hour (integer values from 1 through 24)

- Layer number (integer values >0)

- X cell (integer values >0)

- Y cell (integer values >0)

- Source ID (SMOKE record number, integer values >0)

- Co/St/Cy (Country, state, and county code; values of NSSCCC, with SS and/or CCC replaced by zeros as necessary, depending on the BY option used)

- Country (value is name; optional when country, state, and county code is output)

- State (value is name; optional when country, state, and county code is output)

- County (value is name; optional when country, state, and county code is output)

- SCC (10-character values, with 2 leading zeros for 8-character SCCs)

- SIC (4-character values)

- NAICS (6-character values)

- MACT (6-character values)

- ORIS ID (6-character values)

- Primary Srg (primary surrogate ID code; for area or mobile sources only)

- Fallbk Srg (fallback surrogate ID code; for area or mobile sources only)

- Monthly Prf (monthly temporal profile code)

- Weekly Prf (weekly temporal profile code)

- Diurnal Prf (diurnal temporal profile code)

- Spec Prf (speciation profile code)

- Plant ID (for ORL) or Facility ID (for FF10)

- Char 1 point-source stack ID (for ORL) or ?? (for FF10)

- Char 2 Stack ID (for ORL) or Release Point ID (for FF10)

- Char 3 Segment ID (for ORL) or Process ID (for FF10)

- Char 4 (for point sources only; may or may not be defined)

- Char 5 (for point sources only; may or may not be defined)

- Stk Ht (stack height; for point sources only)

- Stk Dm (stack diameter; for point sources only)

- Stk Tmp (stack exit temperature; for point sources only)

- Stk Vel (stack exit velocity; for point sources only)

- Elevstat (E= elevated, P= Plume-in-grid, L= low-level; for point sources only)

- Plt Name (facility name with quotes around it; for point sources only)

- SCC Description (in quotes)

- SIC Description (in quotes)

- NAICS Description (in quotes)

- MACT Description (in quotes)

- ORIS Description (in quotes)

7.4.1.5.6. Header units

The units used in the header for each column depend on (1) the units in the input files, (2) the units set as output units in the report instructions, and (3) any normalization used. **Smkreport** starts with the units obtained from the input files and application of any matrices of interest. Then all unit conversions that can be made, are made. The conversions are done separately on the numerator and the denominator, but one or both of the conversions will not be done if **Smkreport** has not been configured for the conversions. If this is the case, a note will appear to this effect in the log file. Last, the units are converted to $1/m^2$ if the NORMALIZE CELLAREA instruction has been given.

Below the header units, a line of minus signs will appear in the reports.

7.4.1.5.7. Data rows

The rows of data (i.e., emissions totals or activity totals) are listed sequentially, without spaces between the rows, in sorted order according to the BY instructions, and based on the order listed in the file (just explained above). Data-value-based sorting is not done by **Smkreport**, but rather left for a postprocessing step of other data manipulation software. Unlike the title rows, the length of the data rows constrained only by the maximum number of characters up to 3500 per ASCII line, which varies based on the platform you are using. The ARRANGE instruction can be used to improve the arrangement of data rows when the lines are too large or the program is not working properly because of the large number of data values requested in a report.

The data values in the rows are formatted based on the NUMBER instruction for the report.

The resolution of the data values (e.g., summed by SCC) depends on the BY instructions for the report. The inventory records included in the totals (i.e., which sources are included in the reported emissions) depend on the SELECT instructions for the report. More description of these instructions and how they affect the output is provided with the REPCONFIG file format information given earlier.

7.4.1.6. Elements of the METADATA section

Note: this section is not yet implemented in SMOKE. Once implemented, this section of the reports will begin by a line with the word METADATA on it.

7.4.1.6.1. Input files

In this part of the METADATA section, all of the input files used in running the program are provided, one on each line. The full path and file name will be provided.

7.4.1.6.2. Echoed group definitions

The group definitions from the REPCONFIG file used to generate the file will be listed in this section. The definitions will be merely copied directly from the input file to this section of the output report.

7.4.1.6.3. Echoed report instructions

The report instructions from the REPCONFIG file used for creating each report in the file will be listed in this section. The instructions will be merely copied directly from the input file.

7.4.2. Example reports

Given below are just a few examples of the many types of reports **Smkreport** can generate. Note that no actual emission data values are given in these examples (and sometimes the emission values are not even listed). Also, no example is given with two reports in a single file. When we created these examples, we pretended that the inventory used to generate the reports had only CO and NOx available. Also, we have not provided the entire reports, but rather a sampling of what each report looks like, without showing all states, SCCs, etc.

7.4.2.1. Example 1

```
Country-total point-source inventory emissions, with country names
1996 Inventory
No gridding

Co/St/Cy, Country Name, CO       , NOX
        ,              , [tons/yr], [tons/yr]
---------------------------------------------
  000000, US           , 602390.4 , 302123.1
  100000, Canada       , 312233.3 , 197158.9
  200000, Mexico       , 809822.1 , 289321.3
```

7.4.2.2. Example 2

```
State-total point-source emissions, with state names
1996 Inventory
No gridding
```

```
Co/St/Cy, State Name     , CO        , NOX
        ,                , [tons/yr] , [tons/yr]
-------------------------------------------------
  010000, Alabama        ,    2390.4 ,    2123.1
  037000, North Carolina,    5233.3 ,    2400.9
  045000, South Carolina,    3822.1 ,    1321.3
```

7.4.2.3. Example 3

```
County-total area-source emissions, with state and county names
1996 Inventory
No gridding

Co/St/Cy, State Name     , County Name , CO        , NOX
        ,                ,             , [tons/yr] , [tons/yr]
--------------------------------------------------------------
  037001, North Carolina, Alamance Co , 0.2344E+03, 0.1235E+03
  037003, North Carolina, Alexander Co, 0.1354E+03, 0.9909E+02
  037005, North Carolina, Alleghany Co, 0.9807E+02, 0.7897E+02
  037007, North Carolina, Anson Co    , 0.8734E+03, 0.9807E+02
  037009, North Carolina, Ashe Co     , 0.1140E+03, 0.9871E+02
```

7.4.2.4. Example 4

```
Hourly area-source emissions by county for first 3 counties in NC
1996 Inventory
No gridding
July 1, 1996 at 0 GMT to July 7, 1996 at 23 GMT

Day       , Hour, Co/St/Cy, CO       , NOX
          ,     ,         , [tons/hr] , [tons/hr]
-------------------------------------------------
07/01/1996,   0,   037001,    0.1234,   0.5678
07/01/1996,   0,   037003,    0.1247,   0.5725
07/01/1996,   0,   037005,    0.1293,   0.5548
07/01/1996,   1,   037001,    etc.
07/01/1996,   1,   037003,
07/01/1996,   1,   037005,
07/01/1996,   2,   037001,
07/01/1996,   2,   037003,
07/01/1996,   2,   037005,
```

7.4.2.5. Example 5

In this example, the data are sorted by the first three columns (the BY settings) from left to right. The x-cell cycles the slowest, followed by the y-cell, and then by the county code.

```
Grid-cell inventory area-source emissions by county, daily average
1996 Inventory
Grid name NC36_03X03

X cell, Y cell, Co/St/Cy, CO       , NOX
      ,       ,         , [kg/day], [kg/day]
```

```
      ------------------------------------------------
      1,      1,    037001, 0.123556, 0.0789112
      1,      1,    037003, etc.
      1,      2,    037001,
      1,      2,    037003,
      2,      1,    037001,
```

7.4.2.6. Example 6

```
SCC-total mobile-source emissions by day from hourly data
1996 Inventory
No gridding
July 1, 1996 at 0 GMT to July 7, 1996 at 23 GMT

Date      , SCC        , CO        , NOX
          ,            , [tons/day], [tons/day]
------------------------------------------------
07/01/1996, 2201001110,    123.321,    456.654
07/01/1996, 2201001130,    etc.
07/01/1996, 2201001150,
07/01/1996, 2201001170,
07/01/1996, 2201001190,
07/01/1996, 2201001210,
07/01/1996, 2201001230,
```

7.4.2.7. Example 7

```
SCC-total mobile-source daily-total emissions by state (no state names)
North Carolina and South Carolina
1996 Inventory
No gridding
July 1, 1996 at 0 GMT to July 7, 1996 at 23 GMT

Date      , Co/St/Cy, SCC        , CO        , NOX
          ,         ,            , [tons/day], [tons/day]
--------------------------------------------------------
07/01/1996,    37000, 2201001110,    123.321,    456.654
07/01/1996,    37000, 2201001130,    etc.
07/01/1996,    37000, 2201001150,
07/01/1996,    37000, 2201001170,
07/01/1996,    37000, 2201001190,
07/01/1996,    37000, 2201001210,
07/01/1996,    37000, 2201001230,
07/01/1996,    45000, 2201001110,
07/01/1996,    45000, 2201001130,
07/01/1996,    45000, 2201001150,
07/01/1996,    45000, 2201001170,
07/01/1996,    45000, 2201001190,
07/01/1996,    45000, 2201001210,
07/01/1996,    45000, 2201001230,
```

7.4.2.8. Example 8

```
Elevated hourly emissions by grid cell and point source
Only 2 point sources in inventory (wouldn't that be nice?)
1996 Inventory
Grid name NC36_03X03
July 1, 1996 at 0 GMT to July 7, 1996 at 23 GMT

Date      , Hour, X cell, Y cell, Source ID, CO         , NOX
          ,     ,       ,       ,          , [tons/day], [tons/day]
------------------------------------------------------------------
07/01/1996,   0,     1,     1,         1,   123.321,    456.654
07/01/1996,   0,     1,     1,         2,    12.345,     56.789
07/01/1996,   0,     1,     2,         1,   123.321,    456.654
07/01/1996,   0,     1,     2,         2,    12.345,     56.789
07/01/1996,   0,     1,     3,         1,   etc.
07/01/1996,   0,     1,     3,         2,
07/01/1996,   0,     2,     1,         1,
07/01/1996,   0,     2,     1,         2,
07/01/1996,   0,     2,     2,         1,
07/01/1996,   0,     2,     2,         2,
07/01/1996,   0,     2,     3,         1,
07/01/1996,   0,     2,     3,         2,
07/01/1996,   0,     3,     1,         1,
07/01/1996,   0,     3,     1,         2,
07/01/1996,   0,     3,     2,         1,
07/01/1996,   0,     3,     2,         2,
07/01/1996,   0,     3,     3,         1,
07/01/1996,   0,     3,     3,         2,
07/01/1996,   1,     1,     1,         1,
07/01/1996,   1,     1,     1,         2,
07/01/1996,   1,     1,     2,         1,
07/01/1996,   1,     1,     2,         2,
07/01/1996,   1,     1,     3,         1,
07/01/1996,   1,     1,     3,         2,
```

7.4.2.9. Example 9

```
Breweries and Wineries SCC-total area-source emissions for a subgrid
Grid name: NC36_24X24
Subgrid cells selected:
 (1,1) TO (4,4)
 (6,6) TO (9,9)

X cell, Y cell, SCC       , CO        , NOX
      ,       ,           , [tons/day], [tons/day]
--------------------------------------------------
    1,      1, 2302070001, 0.1234E-03, 0.4567E-04
    1,      1, 2302070005, etc.
    1,      2, 2302070001,
    1,      2, 2302070005,
    1,      3, 2302070001,
    1,      3, 2302070005,
```

```
      1,        4, 2302070001,
      1,        4, 2302070005,
      2,        1, 2302070001,
      2,        1, 2302070005,
```

7.4.2.10. Example 10

```
Grid-cell-normalized area-source daily-total speciated emissions for NC
Grid name: NC36_24X24
Cell-area-normalized emissions using cell size 36000 m X 36000 m
July 1, 1996 at 0 GMT to July 7, 1996 at 23 GMT

Date       , Co/St/Cy, State           , CO           , NO        , NO2
           ,          ,                 , [g/day*m^2], [g/day*m^2], [g/day*m^2]
-------------------------------------------------------------------------
07/01/1996,    37000, North Carolina,   112.3456,       23.3456,      1.1234
07/02/1996,    37000, North Carolina,   112.3456,       23.3456,      1.1234
07/03/1996,    37000, North Carolina,   112.3456,       23.3456,      1.1234
07/04/1996,    37000, North Carolina,   112.3456,       23.3456,      1.1234
07/05/1996,    37000, North Carolina,   112.3456,       23.3456,      1.1234
07/06/1996,    37000, North Carolina,   112.3456,       23.3456,      1.1234
07/07/1996,    37000, North Carolina,   112.3456,       23.3456,      1.1234
```

7.5. Quality Assurance Methods

To determine whether your run worked correctly, you must quality assure your emissions processing. To one degree or another, you must design your quality assurance techniques to be specific to each emissions modeling effort, since no two efforts are exactly the same. This section describes some general methods that can be followed to assist users with quality assuring their emissions processing.

As stated in the introduction to this chapter, quality assurance consists of at least the following four components. These are discussed in Section 7.5.1, "Setting up SMOKE" [336], Section 7.5.2, "Reviewing log files" [337], Section 7.5.3, "Reviewing reports created by SMOKE core programs" [339], and Section 7.5.4, "Reviewing and customizing reports created by **Smkreport**" [340].

- Setting up SMOKE using available documentation (this manual) on file formats and script setup.

- Reviewing log files for errors and warnings, then resolving any problems.

- Reviewing reports generated by SMOKE core programs (e.g., reports from the **Cntlmat** program indicating which controls were applied and the changes made to the inventory emissions).

- Reviewing and customizing reports generated by the **Smkreport** program, and comparing these to one another and to totals from the emission inventory.

With all reporting and analyses, it is important to keep the following principles in mind:

- Focus only on those pollutants and inventories that are key components of air quality problems in your modeling case. For example, it is often not worthwhile analyzing CO emissions, since their contribution to ozone air quality problems is minor. It is often worthwhile to first analyze the raw inventories to see which source categories are contributing the majority of key pollutants, to help target your efforts.

- When performing analysis on a key inventory and pollutant, focus on those SCCs that contribute the majority of the emissions. For example, if 10% of the SCCs contribute 90% of the emissions, focus on quality assuring those 10% of SCCs.

- Keep lists of potential issues, their status, and their resolution. Such lists will be critical to documenting the emissions modeling efforts when you are finished.

- As problems are found, new inventories may be created and analyses rerun. Make sure to use new scenario names in the SMOKE Assigns files when running a new version of the inventory. The `INVEN`, `INVOP`, and `INVID` environment variables can be configured to use the same input directories and new output directories.

- Consider using version-control software (such as the UNIX CVS or SCCS programs) to keep track of changes in your inventories and SMOKE support files.

- If possible, archive the SMOKE configuration, including all inputs and output files, once a set of "final" model-ready emissions is created and used in an air quality model. Inevitably, more changes will be made to the SMOKE setup, but you will most likely want or need to go back to the raw inputs used for a specific case, once it has been run through an AQM.

7.5.1. Setting up SMOKE

The information in Chapter 8, *SMOKE Input Files* [345], which provides detailed documentation on SMOKE input file formats, can be used to help ensure the quality of your modeling efforts. Although SMOKE programs check the input file formats when they are read and used, you should not rely on this capability entirely. There is no substitute for carefully comparing your input files with the file formats in this manual. The files should be checked in the following ways:

- Make sure all required fields specified in the manual are present.

- Make sure all fields are in the correct position in the file.

- Make sure the type of the field (integer, real, character) is consistent between the file and the manual. Usually, it is acceptable to provide an integer value for a real field, if the data warrant this.

- Make sure the file has been converted from DOS/Windows to UNIX format. This is often an issue when users edit their files on a Windows computer and move the file to a UNIX (including Linux) workstation for further processing. Files that have not been transferred properly will have "^M" entries at the end of each line and possibly a "^Z" entry at the end of the file. All of these erroneous fields must be removed. Some options for making this conversion automatically include the **dos2unix** and **to_unix** UNIX command-line programs.

- Make sure tabs are not used in the file. SMOKE does not support tabs as delimiters in the input files, and the readers will not be able to read the data correctly if tabs are present.

Chapter 4, *Using SMOKE Scripts* [117] provides documentation on the SMOKE example scripts. We recommend that new users start with these example scripts to enhance the quality of their experience using SMOKE - and thus the ultimate quality of their emissions modeling results. The script documentation should be reviewed carefully before making changes to the settings in the scripts. In addition, Chapter 5, *SMOKE Utility Tools* [177], Chapter 6, *SMOKE Core Programs* [211], and Section 7.2, "**Smkreport** Program" [305] provide documentation on the UNIX environment variables that the scripts use to control the SMOKE programs. These settings can make a large difference in the emissions modeling results. Users should review each setting in the scripts, understand its purpose, and make an informed decision about the value to use for each setting prior to using the SMOKE run scripts on new inventory data. The end of Chapter 4, *Using SMOKE Scripts* [117] provides some additional information about how to reconfigure

your run scripts; these instructions should be followed carefully to ensure high quality in your emissions modeling.

A useful way to ensure that the settings have all been checked and set correctly is to create a table that includes all SMOKE settings for each source category. This table can then be filled in prior to running SMOKE for a new case, while reviewing the SMOKE documentation in detail for the correct settings. As the scripts are created for a new case, this table can be the reference for setting up the script properly. Prior to running the new script, its settings should be compared to the reference table as a final check that all the correct settings have been used.

7.5.2. Reviewing log files

The SMOKE log files are the first source of information about whether your SMOKE run has completed properly. The simplest thing to check in the log file is that the program ran to completion. If it did, the last line of the log file will be "Normal Completion of program..." followed by the program name. This line is simply a first step to understanding whether your program actually completed successfully because this "successful completion" statement will be printed even if your program ran with warnings. If errors occurred in your program run, this line will not be printed.

To fully evaluate whether your program run was indeed successful, you must review the entire log file and also perform additional QA as described in Section 7.5.3, "Reviewing reports created by SMOKE core programs" [339] and Section 7.5.4, "Reviewing and customizing reports created by **Smkreport**" [340]. In this section, we describe how to further review the log files. Figure 7.1, "Flow diagram for examining log files" [338] is a flow diagram that gives step-by-step instructions on how to evaluate the errors and warnings in the log files.

In the diagram, "SMOKE support" refers to several resources that you can use for support on SMOKE. The primary support mechanism is the Community Modeling and Analysis System (CMAS) Center Help Desk [http://www.cmascenter.org/html/help.html]. This web site can be used to look up questions and answers as well as submit new questions about SMOKE usage and behavior. Another resource available to all emissions modelers is the emregional listserv [http://webmail0.isis.unc.edu:82/lists/index.php?reason=results&list=emregional]. This listserv is hosted by the University of North Carolina at Chapel Hill's Carolina Environmental Program (CEP) and is open to all questions, concerns, and issues related to emissions modeling.

Figure 7.1. Flow diagram for examining log files
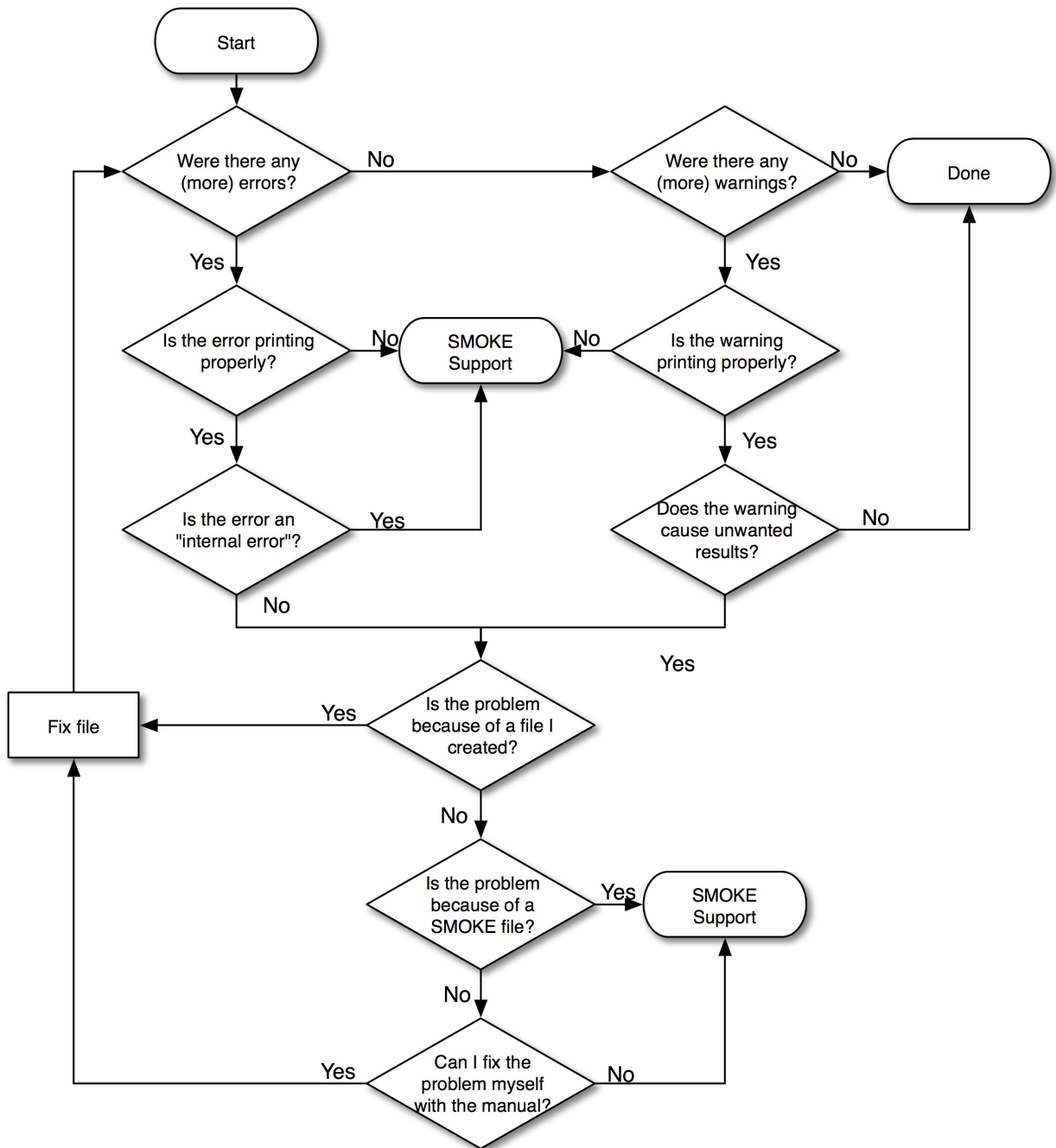
One technique you can use to quickly create a list of the major types of errors or warnings that occurred, without listing all of the many specific instances, is to use the UNIX **grep** command. This approach will find the types of errors and warnings, but when multiple instances of an error or warning are present (e.g., for multiple sources) only one instance of the error or warning will be listed. The UNIX command to perform this function is:

```
grep ERROR <logfile> | cut -c1-50 | sort -u
```

The "ERROR" in this command can be replaced by "WARNING", and the number "50" can be increased or reduced depending on whether you need more or fewer of the warnings in order to obtain the information you want. The "50" controls how many columns of text are included. The **sort -u** performs a unique (-u) sort that will eliminate duplicate records created by the **cut** part of the command. So, since more lines will be similar when the **cut** command includes fewer columns, the unique sort will produce fewer unique entries as the **cut** limit is decreased. The number 50 is usually a good starting point, since it captures the majority of the critical part of the warning, without capturing less significant details such as a line number from an input file on which the error or warning occurred.

In addition to reviewing the log files for errors and warnings, we recommend that you compare the environment variable settings listed in the log files with those you intended to use. Previously, we recommended that you create a table that includes all of the correct SMOKE scripts settings to be used in the modeling effort. The settings shown in the log files should be compared back to that master table to ensure that everything worked properly. This can help you prevent the SMOKE programs from obtaining and using incorrect settings due to typographical errors in the scripts.

Finally, the log files include all of the names of the input and output files. These should be reviewed to confirm that the correct input files and output files were actually used by the program. It is possible that an error in the Assigns file or script could cause the wrong input files to be used, or the output files to be written to the wrong locations.

7.5.3. Reviewing reports created by SMOKE core programs

In addition to log files, some SMOKE programs also create reports. These reports can help provide quality assurance concerning the steps taken by the program. The programs and their associated reports and uses are as follows:

Smkinven: Creates the REPINVEN file, which documents the inventory data read, which data where kept, and the inventory totals for each data value. It also includes the area-to-point conversion information.

Laypoint: Creates the REPRTLAY file, which documents any point sources for which the **Laypoint** program computed a plume rise into a layer greater than the layer number set by the REP_LAYER_MAX setting.

Spcmat: Creates the ASSUP, MSSUP, and PSSUP files, which include the speciation profile codes assigned to each source by pollutant. These files can be used with the ASCII component of the SMOKE intermediate inventory file (AREA, MOBL, and PNTS) to quickly determine the speciation profile assigned to a specific source. These files are also input files to the **Smkreport** program, for creating more involved summaries of emissions and/or sources including the speciation profiles.

Grdmat: Creates the AGSUP and MGSUP files, which include the spatial surrogate codes assigned to each source. They also indicate whether that assignment was a fallback or original assignment. This file can be used with the ASCII component of the SMOKE intermediate inventory file (AREA and MOBL) to quickly determine the spatial surrogate assigned to a specific source. These files can also be used by the **Smkreport** program, for creating more involved summaries of emissions and/or sources including the spatial surrogate codes.

Temporal: Creates the ATSUP, MTSUP, and PTSUP files, which include the monthly, weekly, and diurnal profile codes assigned to each source for each pollutant. These files can be used with the ASCII component of the SMOKE intermediate inventory file (AREA, MOBL, and PNTS) to quickly determine the temporal profiles assigned to a specific source. When no pollutant-specific temporal profile assignments have been made, these files can also be used by the **Smkreport** program, for creating more involved summaries of emissions and/or sources including the temporal profile codes.

Cntlmat: Writes several report files (documented with the **Cntlmat** program in Chapter 6, *SMOKE Core Programs* [211]) that summarize the growth and control factors. Separate reports are written for each type of projection (growth, multiplicative control, reactivity control) with emissions before and after each control. Another report is written that summarizes all projections made to each source.

Smkmerge: Writes a report of state and/or county totals that are created at the same time as the model-ready files. These totals are guaranteed to match the actual state/county totals in the model-ready gridded, hourly outputs (unless there is a problem introduced into **Smkmerge**). These reports are useful even though they duplicate the **Smkreport** reports in some ways, because they allow you to ensure that the **Smkmerge** totals and **Smkreport** totals are the same - indicating that **Smkreport** was run with the correct files. Because **Smkreport** is run separately from **Smkmerge**, this is not necessarily the case, so you should confirm it with the **Smkmerge** report.

7.5.4. Reviewing and customizing reports created by Smkreport

This section provides some ideas on things to check after SMOKE has been run to a successful completion (and after the other QA steps described above have been completed). The review steps listed here will help ensure that SMOKE is working properly. However, these recommendations will not enable you to identify any missing sources (the source exists but was not included in the inventory) or wrong emissions sources (the source is in the inventory, but the emissions are completely incorrect), so it is up to you to understand your emission inventory and make sure that it is representative of the emissions for the regions and years of interest.

7.5.4.1. Area, nonroad mobile, and precomputed emissions for on-road mobile sources

The following checks should be performed to ensure that the area-source and nonroad mobile-source processing was done properly:

1. Compare state and/or county totals among the following processing steps:

    a. Inventory import

    b. Spatial allocation

    c. Temporal allocation for key days (e.g., weekday, Saturday, and Sunday)

    d. Chemical speciation (e.g., add up NO/NO$_2$ and VOC species using the SELECT DATA instruction in **Smkreport** with the S- prefix)

    e. Final merge

    f. Make comparisons for multiple grid resolutions, if applicable.

    The differences and similarities revealed at each stage should be these:

    a. Import totals (after **Smkinven** processing) should be the same as inventory totals outside of SMOKE (if available).

    b. Spatial allocation totals should be the same as the import totals for all states/counties except those on a grid boundary. Comparing totals after spatial allocation among multiple grids should reveal that the emissions totals are the same in all grids for states/counties that are completely within the grids.

c.   Temporal allocation should be different for weekdays, Saturday, and Sunday, and different from the average-weekday or average annual value represented by the import totals. The differences should be in the correct direction: generally, area sources are higher on weekends because of lawn/garden emissions. If you are using an average-day inventory (not an annual inventory), confirm that monthly adjustments have not been made. Check that any holiday days have been processed as directed based on the HOLIDAYS files (e.g., that weekday holidays have been processed as Sundays).

d.   Chemical speciation should match total mass before and after (again, the **Smkreport** SELECT DATA option with the S- prefix is helpful here).

e.   Final merged files should match between the **Smkmerge** and **Smkreport** reports.

2.   Create report of temporal allocation factors used, by SCC. Review profiles assigned to SCCs representing the top 50%-80% of key area/nonroad pollutants for your inventory.

3.   Create report of spatial surrogates used, by SCC. Review spatial surrogates assigned to SCCs representing the top 50%-80% of key area/nonroad pollutants for your inventory.

4.   Create report of speciation profiles used, by SCC. Review speciation profiles assigned to SCCs representing the top 50%-80% of key area/nonroad pollutants for your inventory.

5.   Create reports by state/SCC for key states in your domain (the SELECT REGION statement in **Smkreport** can be used to isolate certain states for this report). Review these reports to ensure that the totals are consistent with expectations. Compare and contrast emissions by SCC among states. Include per-capita reports as well as total emissions reports, and review these to determine if per-capita emissions differences make sense from state to state; this approach can often identify outliers in inventories (caused by improper units, double counting, or out-of-date inventory creation approach).

6.   Create spatial plots (e.g., using VERDI) of average-day emissions for each day (these daily-total emissions can be created by running **Smkmerge** with MRG_TEMPORAL set to N). Daily-total plots can also be created from the model-ready plots using the I/O API utility **m3tproc**. Do for all grids (if applicable) and compare.

7.   Create temporal profile plots (e.g., using VERDI) of hourly, gridded data over the entire episode. Look for consistent patterns on consistent days (e.g., all non-holiday weekdays in a single month should have the same value). Do for all grids (if applicable) and compare. Check for holidays and ensure these were processed as Sundays.

7.5.4.2. Point sources

The following checks should be performed to ensure that the point-sources processing was done properly:

1.   Compare state and/or county totals at the following processing steps for all grids:

a.   Inventory import

b.   Spatial allocation

c.   Temporal allocation for weekday, Saturday, and Sunday

d.   Chemical speciation (e.g., add up NO/NO_2 and VOC species using the SELECT DATA instruction in **Smkreport** with the S- prefix)

e.   Layer fractions

f.   Final merge

The differences and similarities revealed at each stage should be these:

a.   Import totals (after **Smkinven** processing) should be the same as inventory totals outside of SMOKE (if available)

b.   Spatial allocation should be the same as the import totals for all states/counties except those on a grid boundary. Comparing totals after spatial allocation among multiple grids should reveal that the emissions totals are the same in all grids for states/counties that are completely within the grids.

c.   Temporal allocation may or may not be different for weekdays, Saturday, and Sunday, and different from the average-weekday or average annual value represented by the import totals. The differences should be in the correct direction: generally, many of the electric generating facilities (major emissions sources) are the same for all days of the week because the power plants never shut down and their emissions do not depend much on the day of the week. Emissions for major manufacturing plants will often be closed completely on weekends and have no emissions. If using an average-day inventory (not an annual inventory), confirm that monthly adjustments have not been made. Check that any holiday days have been processed as directed based on the HOLIDAYS files (e.g., that weekday holidays have been processed as Sundays).

If using day-specific or hour-specific inventory data, then it will be harder to compare the inventory totals before input to SMOKE to the emissions totals after temporal processing. This is because part of a state/county total when using hour-day- or hour-specific data include these data and part do not, but the inventory totals are often available only for the annual and/or average-day emissions values. This limitation can be rectified by calculating the emissions totals for the day- and hour-specific data outside of SMOKE, and analyzing these sources by SCC instead of only by state/county to separate the analysis of the annual and/or average-day emissions from the analysis of the day- and/or hour-specific emissions.

d.   Chemical speciation should match total mass before and after (again, the **Smkreport** SELECT DATA option with the S- prefix is helpful here).

e.   Emissions totals should match before and after layer fractions for all grids. Note that the layer fractions will be different across grids for sources that are in multiple grids because the meteorology data are different.

f.   Final merged files should match between the **Smkmerge** and **Smkreport** reports.

2.   Create report of temporal allocation factors used, by SCC. Review profiles assigned to SCCs representing the top 50%-80% of key point-source pollutants for your inventory.

3.   Create report of speciation profiles used, by SCC. Review speciation profiles assigned to SCCs representing the top 50%-80% of key point-source pollutants for your inventory.

4.   Create reports by state/plant for key states. These should include the grid cell in which the plant is located, which should be reviewed for correctness. A GIS analysis approach could be helpful for this step, to match the grid cells in the report with an independent assessment of the state/county. Major sources should be reviewed for correct emissions and correct locations.

5.   Create reports with layer fractions for all major power plants in key states. Include grid cell in which the plant is located. Perform a reality-check on the plume rise; large power plants should have high plume rise, but the plumes should still be well distributed throughout the model layers.

6. Create the spatial and temporal profile plots discussed in items 6 and 7 above for area/nonroad processing, except for the spatial plots you need to first condense the I/O API model-ready files to 2-D instead of 3-D files. The I/O API utility **vertot** can be used to sum layers from 3-D files to 2-D files.

7.5.4.3. On-road mobile sources using MOBILE6

The following checks should be performed to ensure that the on-road mobile-source processing was done properly:

1. Compare state and/or county totals at the following processing steps for all grids (if applicable):

   a. Inventory import (VMT only)

   b. Spatial allocation (VMT only)

   c. Compare emissions among all days by states. Compare sample weekday emissions daily-totals with average-day VMT to see if sources with high VMT have high emissions. Calculate ratios between VMT and emissions and look for outliers.

   d. Compare temporalized daily-totals with speciation/temporalized daily-totals of mass emissions

   e. Final merge

   The differences and similarities revealed at each stage should be these:

   a. Import totals (after **Smkinven** processing) should be the same as inventory totals outside of SMOKE (if available)

   b. Spatial allocation should be the same as the import totals for all states/counties except those on a grid boundary. Comparing totals after spatial allocation among multiple grids should reveal that the emissions totals are the same in all grids for states/counties that are completely within the grids.

   c. Temporal allocation should be different for weekdays, Saturday, and Sunday (note: no average-day values are available for on-road mobile-source processing using MOBILE6). The differences should be in the correct direction: generally, mobile sources are lower on weekends. Check any holidays to ensure that Sunday emissions were used.

   d. Chemical speciation should match total mass before and after (the SELECT DATA statement in **Smkreport** with the S- prefix can be helpful here).

   e. Final merged files emissions totals from **Smkmerge** should match the temporal/speciated emissions reports from **Smkreport**.

2. Create report of temporal allocation factors used, by SCC; review assignments.

3. Create report of speciation profiles used, by SCC; review assignments.

4. Create reports by state/SCC for key states. Review these for major sources (e.g., gasoline fueled automobiles on interstates, heavy duty diesel vehicles) and to compare contributions among states to see if these make sense.

5. Create the spatial and temporal profile plots discussed in items 6 and 7 above for area/nonroad processing (Section 7.5.4.1, "Area, nonroad mobile, and precomputed emissions for on-road mobile sources" [340]).

Chapter 8. SMOKE Input Files

Table of Contents

8.1. Overview

This chapter contains format information for all input files used by SMOKE. Chapter 9, *SMOKE Intermediate Files* [461] and Chapter 10, *SMOKE Output Files* [483] provide documentation on SMOKE's intermediate and output file formats.

In this overview section, we first provide a few important notes about file formats (Section 8.1.1, "General notes about file formats" [345], and then discuss where to put SMOKE input files (Section 8.1.2, "Where to put SMOKE input files" [346]). Section 8.1.3, "SMOKE input file list" [346] summarizes all the input files needed by SMOKE, organized by function.

The remaining sections in this chapter are broken down as follows:

- Inventory files used by **Smkinven** (Section 8.2, "Inventory Files" [351])

- Temporal allocation files used by **Temporal** (Section 8.3, "Temporal Allocation Files" [380])

- Spatial allocation files used by **Grdmat**, **Normbeis3**, and **Smkmerge** (Section 8.4, "Spatial Allocation Files" [386])

- Chemical speciation files used by **Spcmat** (Section 8.5, "Chemical speciation" [389])

- Growth and controls files used by **Cntlmat** (Section 8.6, "Growth and Controls Files" [395])

- Special merging files used by **Mrggrid** (Section 8.7, "Special Merging Files" [402])

- Meteorology files used by **Tmpbeis3**, **Met4moves**, and **Laypoint** (Section 8.8, "Meteorology Files" [403])

- Source-specific files for biogenics, mobile, and point source processing (Section 8.9, "Source-Specific files" [406])

- Support files used by various programs (Section 8.10, "Other Support Files" [444])

8.1.1. General notes about file formats

- In the tables used to describe the file format for each file, the "Position" column lists a single number or a range of numbers.

When there is a letter or a single number in the "Position" column, this indicates that the file must be list-directed, and therefore the fields on each line need to be delimited by a space, comma, or semicolon. If there are fields that have embedded spaces, commas, or semicolons in a list-directed file, these entries must be surrounded by single or double quotation marks to prevent SMOKE from treating those characters as delimiters.

When a range of numbers is listed in the "Position" column, ie. (1-3), this indicates that the file is column-specific, which means that the entry must be placed exactly within the columns listed.

- For all cross-reference files (e.g. AGREF, MGREF, GSREF), all of the source-characteristic fields are optional. For example, it is permissible to provide only the SCC for use in matching a profile to a source.

  For list-directed files, if there are source characteristic fields that will not be used (such as country/state/county code) that come *before* the required fields in the file, these must be filled in with a zero or -9. If there are unused fields that come *after* the required fields in the file, they can be left blank.

  For column-specific formats, the optional entries can be left blank or set to -9 when they are not used.

- Nearly all input files support full-line and trailing comments. If the first character on a line is a pound sign (#) then the line is treated as a comment. Any characters after an exclamation point (!) are treated as a comment.

8.1.2. Where to put SMOKE input files

A common difficulty for new users is determining where to put the input files for their case. In SMOKE v2.0 and later, we have reduced the number of places in which you need to keep your raw input files. In Chapter 3, *SMOKE Directory Structure* [97], we provide diagrams of the SMOKE directory structures, which help in visualizing the tree of directories that SMOKE uses. Additionally, Section 4.4, "How to use SMOKE" [125] contains instructions for getting set up for new runs, including the files that need to be created and where they should go.

In summary, SMOKE input files must go into a handful of directories. Multiple directories are needed because in some cases (such as speciation profiles and meteorology data), the data are to be shared across multiple runs. In other cases (such as inventory data), the data are specific to a case and should be stored independently to prevent corrupting other modeling runs. To change the directories used for input (or any other) files in SMOKE, you must modify the Assigns file as described in Section 3.3, "How to change the default directory structure" [115].

8.1.3. SMOKE input file list

In this section, we list all of the raw input files to SMOKE. The files are grouped into tables by functional type:

- Table 8.1, "Inventory File Locations" [347]
- Table 8.2, "Temporal Allocation Input File Locations" [347]
- Table 8.3, "Spatial Allocation Input File Locations" [348]
- Table 8.4, "Chemical Speciation Input File Locations" [348]
- Table 8.5, "Growth and Controls Input File Locations" [348]
- Table 8.6, "Special Merging Input File Locations" [348]
- Table 8.7, "Meteorology Input File Locations" [349]
- Table 8.8, "Biogenic-Specific Input File Locations" [349]
- Table 8.9, "MOVES-Specific Input File Locations" [349]
- Table 8.10, "Point-Specific Input File Locations" [350]

- Table 8.11, "Other Support Input File Locations" [350]

Within each table, the files are listed in alphabetical order. When a file is optional, this is noted in the description column of the table.

Table 8.1. Inventory File Locations

Logical File Name	Program	Source Category	File Description	Location
ARINV	**Smkinven**	A	Stationary area/nonpoint or nonroad raw inventory data	$ARDAT
ARDAY	**Smkinven**	A	Area source raw day-specific inventory data	$ARDAY
ARHOUR	**Smkinven**	A	Area source raw hour-specific inventory data	$ARHOUR
MBINV	**Smkinven**	M	Mobile source raw inventory data	$MBDAT
PTDAY	**Smkinven**	P	Point source raw day-specific inventory data	$PTDAT
PTHOUR	**Smkinven**	P	Point source raw hour-specific inventory data	$PTDAT
PTINV	**Smkinven**	P	Point source raw inventory data	$PTDAT

Table 8.2. Temporal Allocation Input File Locations

Logical File Name	Program	Source Category	File Description	Location
[A|M|P]TPRO	**Temporal**	A, M, P	Temporal profile file	$GE_DAT
[A|M|P]TREF	**Temporal**	A, M, P	Temporal cross-reference file	$GE_DAT
HOLIDAYS	**Temporal**	A, M, P	Holidays file	$GE_DAT
PROCDATES	**Temporal**	A, M, P	Procdates file	$GE_DAT

Table 8.3. Spatial Allocation Input File Locations

Logical File Name	Program	Source Category	File Description	Location
SRGDESC	**Grdmat**	A, M	Surrogate definitions and descriptions	$GE_DAT
Area/Mobile surrogate profiles	**Grdmat, Surgtool**	A, M	Area and/or mobile source spatial surrogates	$SRGPRO_PATH
[A|M]GREF	**Grdmat**	A, M	Area source or mobile gridding cross-reference	$GE_DAT
BELD3_A, BELD3_B, BELD3_TOT	**Beld3to2, Normbeis3**	B	Biogenic gridded landuse for use in SMOKE-BEIS3	$BGDAT
BGPRO	**Smkmerge**	B	Biogenic spatial surrogates for county land use	$GE_DAT
GRIDDESC		A, B, M, P	Grid descriptions	$GE_DAT
>GRIDMASK	Grdmat	A, B, M, P	Gridmask	$ARDAT

Table 8.4. Chemical Speciation Input File Locations

Logical File Name	Program	Source Category	File Description	Location
GSCNV	**Spcmat**	A, M, P	Pollutant-to-pollutant conversion file (e.g., ROG to TOG) (optional)	$GE_DAT
GSPRO	**Cntlmat, Spcmat, Tmpbeis3**	A, B, M, P	Speciation profiles file	$GE_DAT
GSREF	**Spcmat**	A, M, P	Speciation cross-reference file (a.k.a., mapping table)	$GE_DAT

Table 8.5. Growth and Controls Input File Locations

Logical File Name	Program	Source Category	File Description	Location
GCNTL	**Cntlmat, Pktreduc**	A, M, P	Control packets or growth packets file	$STATIC

Table 8.6. Special Merging Input File Locations

Logical File Name	Program	Source Category	File Description	Location
FILELIST	**Mrggrid**	A, B, M, P	List of gridded 2-D and/or 3-D files to be merged	$SCRIPTS/run/

Table 8.7. Meteorology Input File Locations

Logical File Name	Program	Source Category	File Description	Location
GRID_CRO_2D	**Tmpbeis3**	B	Grid 2-D cross-point file (optional)	$METDAT
GRID_CRO_3D	**Laypoint**	P	Grid 3-D cross-point file	$METDAT
MET_CRO_2D	**Tmpbeis3**, **Met4moves**, **Laypoint**, **Metscan**	B, M, P	Meteorology 2-D cross-point file	$METDAT
MET_CRO_3D	**Tmpbeis3**, **Met4moves**, **Laypoint**, **Metscan**	B, M, P	Meteorology 3-D cross-point file (optional for biogenics)	$METDAT
MET_DOT_3D	**Laypoint**	P	Meteorology 3-D dot-point file	$METDAT
METLIST	**Met4moves**	B, M	Listing of meteorology files	$STATIC

Table 8.8. Biogenic-Specific Input File Locations

Logical File Name	Program	Source Category	File Description	Location
B3FAC	**Normbeis3**	B	BEIS3 emissions factors	$BGDAT
BIOSEASON	**Tmpbeis3**	B	Bioseason switch file	$GE_DAT

Table 8.9. MOVES-Specific Input File Locations

Logical File Name	Program	Source Category	File Description	Location
MCXREF	**Movesmrg**, **Met4moves**	M	MOVES source county cross-reference	$MBDAT
MEPROC	**Movesmrg**, **Spcmat**	M	Mobile emission processes (e.g., exhaust, evaporative, tire/break wear) with associated activities and pollutants	$MBDAT
MFMREF	**Movesmrg**, **Met4moves**	M	MOVES source fuel month cross-reference	$MBDAT
MRCLIST	**Movesmrg**	M	List of MOVES lookup tables	$MBDAT
SPDPRO	**Movesmrg**	M	Weekday/Weekend hourly speed profiles	$MBDAT

Table 8.10. Point-Specific Input File Locations

Logical File Name	Program	Source Category	File Description	Location
PELVCONFIG	**Elevpoint**	P	Elevated source and PinG selection configuration file	$PTDAT
PSTK	**Smkinven**	P	Replacement stack parameters for invalid stacks	$GE_DAT
PTMPLIST	**Elevpoint**	P	List of PTMP files for **Elevpoint** to use to compute maximum daily emissions	$PTDAT

Table 8.11. Other Support Input File Locations

Logical File Name	Program	Source Category	File Description	Location
ARTOPNT	**Smkinven**	A	Area-to-point conversion file	$INVDIR/other/
COSTCY	**Smkinven**, **Grwinven**, **Temporal**, **Smkmerge**, **Smkreport**	A, B, M, P	Country, state, and county names and time zones	$GE_DAT
GEO-CODE_LEVEL[1-4]	**Smkinven**, **Grwinven**, **Temporal**, **Smkmerge**, **Smkreport**	A, B, M, P	Geographical level codes and descriptions for gridded inventories, only when USE_EXP_GEO_CODES is set to Y, (COSTCY is ignored when GEO-CODE_LEVEL[1-4] files are used)	$GE_DAT
INVTABLE	**Smkinven**, **Movesmrg**, **Spcmat**, **Temporal**, **Smkmerge**	A, M, P	Pollutant and activity names and codes in order of output	$INVDIR/other/
MACTDESC	**Smkreport**	A, P	MACT descriptions	$GE_DAT
NAICSDESC	**Smkreport**	A, P	NAICS descriptions	$GE_DAT
NHAPEX-CLUDE	**Smkinven**	A, M, P	NONHAP exclusions file	$INVDIR/other/
ORISDESC	**Smkinven**	P	ORIS facility descriptions, needed for getting ORIS facility names for reporting purposes	$GE_DAT
SCCDESC	**Smkinven**, **Smkreport**	A, M, P	SCC descriptions	$GE_DAT
SICDESC	**Smkreport**	A, P	SIC descriptions	$GE_DAT
SOURCE_GROUPS	**Movesmrg**, **Smkreport**	A, B, M, P	Source apportionment groups	$INVDIR/other/
VNAMMAP	**Smk2emis**	A, B, M, P	Species name mapping file for **Smk2emis**	$GE_DAT

8.2. Inventory Files

The directories where the inventory files should be stored are listed in Table 8.1, "Inventory File Locations" [347]. The table also lists the programs that use each input file.

8.2.1. Overview

8.2.1.1. Inventory list files

Area-, mobile-, and point-source inventories are provided to SMOKE as ASCII text files using the logical file names `ARINV`, `MBINV`, and `PTINV`, respectively. These files may either contain the actual inventory data (i.e. source characteristics, emissions data, and other information) or they can contain a list of file names with full paths (i.e. `/apps/smoke/data/inventory/nei02/area/arinv.txt`); in the latter case, we call the file a "list file". When the inventory is a list file, SMOKE will open and read the data in each of the listed files and combine the data into one inventory. You may also choose to separate your data into multiple files, splitting the data by country or into EGU versus nonEGU point sources, or to reduce the individual file sizes. Some inventory formats require that a list file is used; this will be noted in the format descriptions.

An inventory list file can be created using the UNIX **ls**. The following command will create the `ARINV` file containing a list of all files starting with `arinv` in the `$ARDAT` directory, complete with full paths.

ls $ARDAT/arinv* > $ARINV

You will still need to edit this file to add format information (see Section 8.2.1.2, "File format header" [351]).

8.2.1.2. File format header

All area-, mobile-, and point-source inventory files must include information specifying the file format being used. This is accomplished using a header entry, either in each individual inventory file or in the list file, if one is being used. Valid format options are:

- #CEM indicates that the file is in CEM format

- #LIST indicates that the file is a list file

- #LIST GRID indicates that the file is a list of global gridded emission inventory files

- #FF10 indicates that the file is in FF10 (Flat File 10) format is shared between different source categories for annual/monthly or daily/hourly and point/nonpoint/nonroad/onroad data. The user specifies the data provided using this command in the header: FF10_[HOURLY|DAILY]_[NONPOINT|POINT|NONROAD|ONROAD].

- #ORL indicates that the file is in ORL format (usually used for toxics inventories)

- #ORL NONPOINT indicates that the file is in ORL format for nonpoint sources

- #ORL FIRE indicates that the file is in ORL format for fires sources

- #ORL FIREEMIS indicates that the file is in ORL format for day-specific fires sources

8.2.1.3. Required and optional fields

Inventory files can include both required and optional fields. In the format tables for each file, required fields are labeled "required"; optional fields are labeled either "recommended", "optional", or "not used by SMOKE". For column-specific file formats, you may leave blank the fields that are optional. List-directed files, on the other hand, must have a default value (usually -9) for these fields.

8.2.1.4. Header records

The ORL and FF10 formats used for the area-, mobile-, and point-source inventories require a set of header records at the beginning of each file. The general format of the header records is:

command parm1 parm2 ... parm *n*

The *command* field starts in the first column of the row. The remainder of the line is list-directed, with each field separated by blanks. The specific header commands vary by source category and format and are listed in Table 8.12, "Inventory header commands for ORL, and FF10 formats" [352].

Table 8.12. Inventory header commands for ORL, and FF10 formats

Command	Parameter	Description and Values
#FF10_ACTIVITY #FF10_[NONPOINT|POINT|NON-ROAD|ONROAD] #FF10_[HOURLY|DAILY]_[NON-POINT|POINT|NONROAD|ON-ROAD] #ORL #ORL NONPOINT #ORL FIRE #ORL FIREEMIS	N/A	N/A
#TYPE	File type	First nonblank character after #TYPE to the last character of the line. For on-road mobile activity data, the word "activity" must appear in this header entry; otherwise, SMOKE assumes that the file contains emissions data.
#COUNTRY	Country name	Valid names are those in the `COSTCY` or `GEOCODE_LEVEL[1-4]`(FF10 Format only) file(s) (US, CANADA, MEXICO, etc.)
#YEAR	Data year	Number from 1900 to 2200
#DESC	Description	Description of the data (not used by SMOKE)
#DATA or #POLID	data1 data2 ... data *n*	In wild/Rx fires ORL format files (#ORL FIRE ("Section 8.2.8.3, "ORL FIRE Format" [380]") and day-specific #ORL FIREEMIS file ("Section 8.2.6.2, "ORL FIRE Emission Format" [368]"), it defines a complete list of pollutant names from day-specific ORL FIREEMIS files; "data1" is the first pollutant name or CAS number, "data2" is the second pollutant name(or CAS number), and so on.

Sample header records are given for each file in their respective sections.

8.2.2. ARDAY: Area source day-specific emissions

8.2.2.1. FF10 Format

The user specifies the data provided using this command in the header: FF10_DAILY_[NON-POINT|POINT|NONROAD|ONROAD].

This FF10 format uses the header described in Section 8.2.1.4, "Header records" [352]. Sample header records for daily nonpoint sources are shown below:

```
#FORMAT    FF10_DAILY_NONPOINT
#COUNTRY   US
#YEAR      2005
#DESC      Nonpoint Daily Source Inventory
#DESC      FF10 Nonpoint Daily format
```

Table 8.13. FF10 Format for individual area day-specific data files

Position	Name	Type	Description
1	COUNTRY	Char(3)	Country Code: e.g., US (required)
2	FIPS	Char(5)	FIPS code for state and county (required)
3	TRI-BAL_CODE	Char(3)	Tribal code (optional; recommended)
4	CENSUS_TRACT	Char	Census tract ID (optional; not currently used by SMOKE)
5	SHAPE_ID	Char	Shape ID (optional; not currently used by SMOKE
6	N/A (un-used)	N/A	Leave blank (not currently used by SMOKE)
7	EMIS_TYPE	Char(3)	Emissions type [ex: EXH, EVP, and so on] (optional; not currently used by SMOKE)
8	SCC	Char(20)	SCC (required)
9	POLL	Char(16)	Pollutant Code (required)
10	OP_TYPE_CD	Char(20)	Operating type code for startup/shutdown/upset (optional; not currently used by SMOKE)
11	CALC_METH-OD	Int	Calculation Method (optional; not currently used by SMOKE)
12	DATE_UP-DATED	Int	Updated date in format of YYYYMMDD (optional; not currently used by SMOKE)
13	MONTH	Int	Month of inventory (required)
14	MONTHTOT	Real	Monthly Total (short tons/month) (required)
15	DAYVAL1	Real	Daily emissions for day 1 (short tons/day) (required)
16	DAYVAL2	Real	Daily emissions for day 2 (short tons/day) (required)
17	DAYVAL3	Real	Daily emissions for day 3 (short tons/day) (required)
	..		..
45	DAYVAL31	Real	Daily emissions for day 31 (short tons/day) (required)
46	COMMENT	Char	(optional)

8.2.3. ARHOUR: Area source hour-specific emissions

8.2.3.1. FF10 Format

The user specifies the data provided using this command in the header: FF10_HOURLY_[NON-POINT|POINT|NONPOINT|ONROAD].

This FF10 format uses the header described in Section 8.2.1.4, "Header records" [352]. Sample header records for daily nonpoint sources are shown below:

```
#FORMAT    FF10_HOURLY_NONPOINT
#COUNTRY   US
#YEAR      2005
#DESC      Nonpoint Hourly Source Inventory
#DESC      FF10 Nonpoint Hourly format
```

Table 8.14. FF10 Format for individual area hour-specific files

Position	Name	Type	Description
1	COUNTRY	Char(3)	Country code: e.g., US (required)
2	FIPS	Char(5)	FIPS code for state and county (required)
3	TRI-BAL_CODE	Char(3)	Tribal code (optional; recommended)
4	CENSUS_TRACT	Char	Census tract ID (optional; not currently used by SMOKE
5	SHAPE_ID	Char	Shape ID (optional; not currently used by SMOKE
6	N/A (un-used)		Leave blank (Not currently used by SMOKE)
7	EMIS_TYPE	Char(3)	Emissions type [ex: EXH, EVP, and so on] (optional; not currently used by SMOKE)
8	SCC	Char(20)	SCC (required)
9	POLL	Char(16)	Pollutant Code (required)
10	OP_TYPE_CD	Char(20)	Operating type code for startup/shutdown/upset (optional; not currently used by SMOKE)
11	CALC_METH-OD	Int	CALC_METHOD (optional; not currently used by SMOKE)
12	DATE_UP-DATED	Int	Updated date in format of YYYYMMDD (optional; not currently used by SMOKE)
13	DATE	Int	Date of inventory in format of YYYYMMDD (required)
14	DAYTOT	Real	Daily Total (short tons/day) (required if SMKINVEN_MONTH is set to 0)
15	HRVAL0	Real	Hourly emissions for hour 0 (short tons/hour)(required)
16	HRVAL1	Real	Hourly emissions for hour 1 (short tons/hour) (required)
17	HRVAL2	Real	Hourly emissions for hour 2 (short tons/hour) (required)
18	HRVAL3	Real	Hourly emissions for hour 3 (short tons/hour) (required)
	..		..
38	HRVAL23	Real	Hourly emissions for hour 23 (short tons/hour) (required)
39	COMMENT	Char	(optional)

8.2.4. ARINV: Area-source annual or average-day emissions

The ARINV file is the input area source inventory file for annual or average-day data. This file can be in FF10 or ORL Formats.

8.2.4.1. FF10 Format

This FF10 format uses the header described in Section 8.2.1.4, "Header records" [352].

For area nonpoint, nonroad and onroad mobile sources annual or monthly inventory data the user specifies the data provided using this command in the header: FF10_[NONPOINT|NONROAD|ONROAD]

Sample header records for nonpoint mobile sources:

```
#FORMAT    FF10_NONPOINT
#COUNTRY   US
#YEAR      2005
#DESC      Nonpoint Source Inventory
#DESC      FF10 Nonpoint format
```

Sample header records for nonroad mobile sources:

```
#FORMAT    FF10_NONROAD
#COUNTRY   US
#YEAR      2005
#DESC      Nonroad Source Inventory
#DESC      FF10 Nonroad format
```

Sample header records for onroad mobile sources:

```
#FORMAT    FF10_ONROAD
#COUNTRY   US
#YEAR      2005
#DESC      Onroad Mobile Source Inventory
#DESC      FF10 Onroad format
```

The FF10 format for area nonpoint, nonroad and onroad mobiles sources is list-directed and shown in Table 8.15, "FF10 Format for ARINV" [357].

Table 8.15. FF10 Format for `ARINV`

Position	Name	Type	Description
1	COUNTRY	Char(3)	Country code; default to "US" (required)
2	FIPS	Char(5)	FIPS code for state and county (required)
3	TRIBAL CODE	Char(3)	Tribal code (optional; recommended)
4	CENSUS_TRACT	Char	Census tract ID (optional; not currently used by SMOKE
5	SHAPE_ID	Char	Shape ID (optional; not currently used by SMOKE
6	SCC	Char(20)	SCC (required)
7	EMIS_TYPE	Char(3)	Emissions type [ex: EXH, EVP, and so on] (optional; not currently used by SMOKE)
8	POLL	Char(16)	Pollutant Code (required)
9	ANN_VALUE	Real	Annual emissions (short tons/year) (required if `SMKINVEN_MONTH` is set to 0)
10	ANN_PCT_RED	Real	Percentage Control Measure Reduction Efficiency (optional)
11	CONTROL_IDS	Char	Control Measure Code (optional; not currently used by SMOKE)
12	CONTROL_MEASURES	Char	Control Measure Values (optional; not currently used by SMOKE)
13	CURRENT_COST	Real	Control Cost (optional; not currently used by SMOKE)
14	CUMULATIVE_COST	Real	Cumulative Cost (optional; not currently used by SMOKE)
15	PROJECTION_FACTOR	Real	Projection factor (optional; not currently used by SMOKE)
16	REG_CODES	Char	Regulatory Code (optional; not currently used by SMOKE)
17	CALC_METHOD	Char	Emission Calculation Method Code (optional; not currently used by SMOKE)
18	CALC_YEAR	Int	Emission Calculation Data Year (optional; not currently used by SMOKE)
19	DATE_UPDATED	Int	Last time updated date in format of YYYYMMDD (optional; not currently used by SMOKE)
20	DATA_SET_ID	Int	Data set identifier (optional; not currently used by SMOKE)
21	JAN_VALUE	Real	Filled in only if data were provided for January in unit of short tons/month (required if `SMKINVEN_MONTH` is set to 1)
...	...	...	...
32	DEC_VALUE	Real	Filled in only if data were provided for December in unit of short tons/month (required if `SMKINVEN_MONTH` is set to 12)

Position	Name	Type	Description
33	JAN_PCTRED	Real	Month-specific percent reduction. Filled in only if data were provided for January in unit of % (optional; not currently used by SMOKE)
...	...	...	...
44	DEC_PCTRED	Real	Month-specific percent reduction. Filled in only if data were provided for December in unit of % (optional; not currently used by SMOKE)
45	COMMENT	Char	Optional; not currently used by SMOKE

8.2.4.2. ORL Formats

Because area sources encompass both nonpoint and nonroad mobile sources, two different ORL formats are available depending on source type. Both formats use the header described in Section 8.2.1.4, "Header records" [352]. Sample header records for nonpoint sources are shown below:

8.2.4.2.1. ORL Nonpoint Format

The ORL nonpoint format is list-directed and shown in Table 8.16, "ORL Nonpoint Format for ARINV" [359].

```
#ORL NONPOINT
#TYPE     Nonpoint Source Toxics Inventory
#COUNTRY  US
#YEAR     1999
#DESC     Draft 1999 version 2.0
```

Table 8.16. ORL Nonpoint Format for `ARINV`

Position	Name	Type	Description
1	FIPS	Char(5)	FIPS code for state and county (required)
2	SCC	Char(20)	Source Classification Code (required)
3	SIC	Char(20)	Standard Industrial Code (recommended)
4	MACT	Char(6)	Maximum Available Control Technology Code (6 characters maximum) (recommended)
5	SRCTYPE	Char(2)	Source type (2 characaters maximum); used by SMOKE in determining applicable MACT-based controls (required) • 01 = major • 02 = Section 12 area source • 03 = nonroad • 04 = onroad
6	NAICS	Char(6)	North American Industrial Classification System Code (6 characters maximum) (optional)
7	POLL	Char(16)	Pollutant CAS number or other name (16 characters maximum) (required; this is called the pollutant code in the NIF)
8	ANN_EMIS	Real	Annual emissions (short tons/year) (required)
9	AVD_EMIS	Real	Average-day emissions (short tons/average day) (optional)
10	CEFF	Real	Control Efficiency percentage (give value of 0-100) (optional; if missing will result in SMOKE default of 0)
11	REFF	Real	Rule Effectiveness percentage (give value of 0-100) (optional; if missing will result in SMOKE default of 100)
12	RPEN	Real	Rule Penetration percentage (give value of 0-100) (optional; if missing will result in SMOKE default of 100)
13	PRIMARY_DEVICE_CODE	Char	Primary Control Equipment Code (optional; not currently used by SMOKE)
14	SECONDARY_DEVICE_CODE	Char	Secondary Control Equipment Code (optional; not currently used by SMOKE)
15	DATA_SOURCE	Char	Flag indicating the source of the data (e.g., state submitted or EPA-computed) (optional; not currently used by SMOKE)
16	YEAR	Char	Year of the emissions (optional; not currently used by SMOKE)
17	TRIBAL_CODE	Char(3)	Indicates the particular tribe that submitted the data (optional; recommended)
18	MACT_FLAG	Char	MACT flag indicating the relationship of the MACT rule and the particular source (15 characters maximum)(optional; not currently used by SMOKE)

Position	Name	Type	Description
19	PRO- CESS_MACT_COM- PLI- ANCE_STATUS	Char	Indicate the relationship of the MACT rule and the partic- ular source (optional; not currently used by SMOKE)
20	START_DATE	Char(8)	Indicates originally-reported start date prior to annualizing emissions, which may be used for building temporal pro- files for specific sources (8 characters maximum) (option- al; not currently used by SMOKE)
21	END_DATE	Char(8)	Indicates originally-reported end date prior to annualizing emissions, which may be used for building temporal pro- files for specific sources (8 characters maximum) (option- al; not currently used by SMOKE)
22	WINTER_THROUGH- PUT_PCT	Real	Percentage throughout in winter (optional; not currently used by SMOKE)
23	SPRING_THROUGH- PUT_PCT	Real	Percentage throughout in spring (optional; not currently used by SMOKE)
24	SUM- MER_THROUGH- PUT_PCT	Real	Percentage throughout in summer(optional; not currently used by SMOKE)
25	FALL_THROUGH- PUT_PCT	Real	Percentage throughout in fall (optional; not currently used by SMOKE)
26	ANNU- AL_AVG_DAYS_PER_WEEK	Real	Annual average days of operation per week (optional; not currently used by SMOKE)
27	ANNU- AL_AVG_WEEKS_PER_YEAR	Real	Annual average weeks of operation per year (optional; not currently used by SMOKE)
28	ANNU- AL_AVG_HOURS_PER_DAY	Real	Annual average hours of operation per day (optional; not currently used by SMOKE)
29	ANNU- AL_AVG_HOURS_PER_YEAR	Real	Annual average hours of operation per year (optional; not currently used by SMOKE)
30	PERI- OD_DAYS_PER_WEEK	Real	Days per week operating during the period (optional; not currently used by SMOKE)
31	PERI- OD_WEEKS_PER_PERI- OD	Real	Weeks per period operating during the period (optional; not currently used by SMOKE)
32	PERI- OD_HOURS_PER_DAY	Real	Hours of operation per day during the period (optional; not currently used by SMOKE)
33	PERI- OD_HOURS_PER_PERI- OD	Real	Hours of operation per period during the period(optional; not currently used by SMOKE)
34	CON- TROL_MEAS- URES	Char	An & delimited list of control measure abbreviations (optional; not currently used by SMOKE)
35	PCT_REDUC- TION	Char	An & delimited list of percent reductions for the applied control measures (optional; not currently used by SMOKE)

Position	Name	Type	Description
36	CUR-RENT_COST	Char	The cost of the most recently applied control strategy for this source (optional; not currently used by SMOKE)
37	CUMULAT-IVE_COST	Char	The cumulative cost for all applied control measures (optional; not currently used by SMOKE)

8.2.4.2.2. ORL Nonroad Format

The ORL nonroad format is list-directed and shown in Table 8.17, "ORL Nonroad Format for ARINV" [362].

```
#ORL
#TYPE     Nonroad Source Toxics Inventory
#COUNTRY  US
#YEAR     1999
#DESC     Draft 1999 version 2.0
```

Table 8.17. ORL Nonroad Format for `ARINV`

Position	Name	Type	Description
1	FIPS	Char(5)	FIPS code for state and county (required)
2	SCC	Char(20)	Source Classification Code (required)
3	POLL	Char(16)	Pollutant CAS number or other name (required; this is called the pollutant code in the NIF)
4	ANN_EMIS	Real	Annual emissions (short tons/year) (required)
5	AVD_EMIS	Real	Average-day emissions (short tons/average day) (optional)
6	CEFF	Real	Control Efficiency percentage (give value of 0-100) (optional; if missing will result in SMOKE default of 0)
7	REFF	Real	Rule Effectiveness percentage (give value of 0-100) (optional; if missing will result in SMOKE default of 100)
8	RPEN	Real	Rule Penetration percentage (give value of 0-100) (optional; if missing will result in SMOKE default of 100)
9	SRCTYPE	Char(2)	Source type ; used by SMOKE in determining applicable MACT-based controls (required) • 01 = major • 02 = Section 12 area source • 03 = nonroad • 04 = onroad
10	DATA_SOURCE	Char	Flag indicating the source of the data (e.g., state submitted or EPA-computed) (optional; not currently used by SMOKE)
11	YEAR	Char	Year of the emissions (optional; not currently used by SMOKE)
12	TRIBAL_CODE	Char(3)	Indicates the particular tribe that submitted the data (optional; recommended)
13	START_DATE	Char(8)	Indicates originally-reported start date prior to annualizing emissions, which may be used for building temporal profiles for specific sources (optional; not currently used by SMOKE)
14	END_DATE	Char(8)	Indicates originally-reported end date prior to annualizing emissions, which may be used for building temporal profiles for specific sources (optional; not currently used by SMOKE)
15	WINTER_THROUGHPUT_PCT	Real	Percentage throughout in winter (optional; not currently used by SMOKE)
16	SPRING_THROUGHPUT_PCT	Real	Percentage throughout in spring (optional; not currently used by SMOKE)
17	SUMMER_THROUGHPUT_PCT	Real	Percentage throughout in summer (optional; not currently used by SMOKE)

Position	Name	Type	Description
18	FALL_THROUGH-PUT_PCT	Real	Percentage throughout in fall (optional; not currently used by SMOKE)
19	ANNU-ALAVGDAYSPERWEEK	Real	Annual average days of operation per week (optional; not currently used by SMOKE)
20	ANNU-ALAVGWEEKSPERYEAR	Real	Annual average weeks of operation per year (optional; not currently used by SMOKE)
21	ANNU-ALAVGHOURSPERDAY	Real	Annual average hours of operation per day (optional; not currently used by SMOKE)
22	ANNU-ALAVGHOURSPERYEAR	Real	Annual average hours of operation per year (optional; not currently used by SMOKE)
23	PERI-ODDAYSPERWEEK	Real	Days per week operating during the period (optional; not currently used by SMOKE)
24	PERI-ODWEEKSPERPERI-OD	Real	Weeks per period operating during the period (optional; not currently used by SMOKE)
25	PERI-ODHOURSPERDAY	Real	Hours of operation per day during the period (optional; not currently used by SMOKE)
26	PERI-ODHOURSPERPERI-OD	Real	Hours of operation per period during the period(optional; not currently used by SMOKE)
27	CON-TROL_MEAS-URES	Char	An & delimited list of control measure abbreviations (optional; not currently used by SMOKE)
28	PCT_REDUC-TION	Char	An & delimited list of percent reductions for the applied control measures (optional; not currently used by SMOKE)
29	CUR-RENT_COST	Char	The cost of the most recently applied control strategy for this source (optional; not currently used by SMOKE)
30	CUMULAT-IVE_COST	Char	The cumulative cost for all applied control measures (optional; not currently used by SMOKE)

8.2.5. `MBINV`: Mobile-source activity and emissions

The `MBINV` file is the input mobile-source inventory data file. Two formats are available: one for activity data in FF10 format and one for emissions data in ORL format.

8.2.5.1. ORL Format

The ORL format is list-directed and uses the header described in Section 8.2.1.4, "Header records" [352]. Sample header records are shown below:

```
#ORL
#TYPE       Mobile Source Toxics Inventory
#COUNTRY    US
#YEAR       1999
#DESC       Draft 1999 version 2.0
```

The format for the data records is shown in Table 8.18, "ORL Format for `MBINV`" [364].

Table 8.18. ORL Format for `MBINV`

Position	Name	Type	Description
1	FIPS	Char(5)	FIPS code for state and county (required)
2	SCC	Char(20)	Source Classification Code (required)
3	POLL	Char(16)	Pollutant CAS number or other name (required; this is called the pollutant code in the NIF)
4	ANN_EMIS	Real	Annual emissions (short tons/year) (required)
5	AVD_EMIS	Real	Average-day emissions (short tons/average day) (optional)
6	SRCTYPE	Char(2)	Source type; used by SMOKE in determining applicable MACT-based controls (required) • 01 = major • 02 = Section 12 area source • 03 = nonroad • 04 = onroad
7	DATA_SOURCE	Char	Flag indicating the source of the data (e.g., state submitted or EPA-computed) (optional; not currently used by SMOKE)
8	YEAR	Char	Year of the emissions (optional; not currently used by SMOKE)
9	TRIBAL_CODE	Char(3)	Indicates the particular tribe that submitted the data (optional; recommended)
10	CEFF	Real	Control Efficiency percentage (give value of 0-100) (optional; not currently used by SMOKE)
11	REFF	Real	Rule Effectiveness percentage (give value of 0-100) (optional; not currently used by SMOKE)
12	RPEN	Real	Rule Penetration percentage (give value of 0-100) (optional; not currently used by SMOKE)
13	CONTROL_MEASURES	Char	An & delimited list of control measure abbreviations (optional; not currently used by SMOKE)
14	PCT_REDUCTION	Char	An & delimited list of percent reductions for the applied control measures (optional; not currently used by SMOKE)
15	CURRENT_COST	Char	The cost of the most recently applied control strategy for this source (optional; not currently used by SMOKE)
16	CUMULATIVE_COST	Char	The cumulative cost for all applied control measures (optional; not currently used by SMOKE)

8.2.5.2. FF10 Activity Inventory Format

The FF10 format is list-directed and uses the header described in Section 8.2.1.4, "Header records" [352]. Sample header records are shown below:

```
#FORMAT   FF10_Activity
#COUNTRY  US
#YEAR     2008
#DESC     Annual Mobile Source Source Inventory
#DESC     FF10
```

The format for the data records is shown in Table 8.19, "FF10 Format for MBINV" [366].

Table 8.19. FF10 Format for `MBINV`

Position	Name	Type	Required?	Description
1	COUNTRY	Char(3)	Y	Country code; default to "US"
2	FIPS	Char(5)	Y	FIPS code for state and county
3	TRIBAL CODE	Char(3)	N	Indicates the particular tribe that submitted the data (optional; not currently used by SMOKE)
4	CENSUS_TRACT	Char(20)	N	Not currently used by SMOKE
5	SHAPE_ID	Char(20)	N	Not currently used by SMOKE
6	SCC	Char(20)	Y	SCC
7	ACT_PARM_TYPE	Char(10)	N	Not currently used by SMOKE
8	ACT_PARM_UOFMSR	Char(20)	N	
9	ACTIVITY_TYPE	Char(16)	Y	Activity variable names are VMT, VPOP, and SPEED
10	ANN_PARM_VALUE	Real	Y	Annual activity values in unit of miles/yr for annual VMT and miles/hr for SPEED
11	CALC_YEAR	Int	N	Year of inventory
12	DATE_UPDATED	Int	N	Updated Date in fomrat of YYYYMMDD
13	DATA_SET_ID	Int	N	Not currently used by SMOKE
14	JAN_VALUE	Real	N	Filled in only if data were provided for January in unit of miles/month for VMT and miles/hr for SPEED
...	...	...	...	...
27	DEC_VALUE	Real	N	Filled in only if data were provided for December in unit of miles/month for VMT and miles/hr for SPEED
28	COMMENT	Char	N	Optional; not currently used by SMOKE

8.2.6. `PTDAY`: Point source day-specific emissions

The `PTDAY` file is the input point-source inventory file for day-specific data. The file must be a list file (see Section 8.2.1.1, "Inventory list files" [351]) and the files listed in the `PTDAY` file must be in the format shown in Table 8.20, "FF10 Format for individual point day-specific data files" [368] and Table 8.21, "Format for individual day-specific data files for fires : ORL FIREEMIS" [369]. The contents of the day-specific data files vary depending on whether the `PTINV` file is in FF10 or ORL FIRE format. The day-specific files can have a #COUNTRY header entry to set the country code (see Section 8.2.1.4, "Header records" [352]). If no #COUNTRY header is used, then SMOKE will use the default country code of 0 (which corresponds to the United States in the default `COSTCY` file).

8.2.6.1. FF10 Format

This FF10 format uses the header described in Section 8.2.1.4, "Header records" [352]. For daily or hourly specific emissions, the user specifies the data provided using this command in the header: FF10_DAILY_POINT.

Sample header records for daily point source data are shown below:

```
#FORMAT    FF10_DAILY_POINT
#COUNTRY   US
#YEAR      2005
#DESC      Point Daily Source Inventory
#DESC      FF10 Point Daily format
```

Table 8.20. FF10 Format for individual point day-specific data files

Posi- tion	Name	Type	Description
1	COUNTRY	Char(3)	Country code: e.g., US (required)
2	FIPS	Char(5)	FIPS code for state and county (required)
3	TRIBAL_CODE	Char(3)	Tribal code (optional; recommended)
4	FACILITY_ID	Char(15)	Facility Identification Code (required; this is the same as the Plant ID in the ORL Format)
5	UNIT_ID	Char(15)	Unit Identification Code (required; this is the same as the Point ID in the ORL Format) (required)
6	REL_POINT_ID	Char(15)	Release Point Identification Code (recommended; this is the same as the Stack ID in the ORL Format)
7	PROCESS_ID	Char(15)	Process Identification Code characters (recommended; this is the same as the Segment ID in the ORL Format)
8	SCC	Char(20)	SCC (required)
9	POLL	Char(16)	Pollutant Code (required)
10	OP_TYPE_CD	Char(20)	Operating type code for startup/shutdown/upset (optional; not currently used by SMOKE)
11	CALC_METH- OD	Int	CALC_METHOD (optional; not currently used by SMOKE)
12	DATE_UP- DATED	Int	Updated date in format of YYYYMMDD (optional; not currently used by SMOKE)
13	MONTH	Int	Month of inventory (required)
14	MONTHTOT	Real	Monthly Total (short tons/month) (required)
15	DAYVAL1	Real	Daily emissions for day 1 (short tons/day) (required)
16	DAYVAL2	Real	Daily emissions for day 2 (short tons/day) (required)
17	DAYVAL3	Real	Daily emissions for day 3 (short tons/day) (required)
	..		..
45	DAYVAL31	Real	Daily emissions for day 31 (short tons/day) (required)
46	COMMENT	Char	(optional)

8.2.6.2. ORL FIRE Emission Format

The SMOKE requires that the sources in the day-specific fires inventory format (#ORL FIREEMIS) PTDAY file also be in the PTINV file described in Section 8.2.8.3, "ORL FIRE Format" [380]. The values of the fields that are common to the day-specific data and the master (annual) data from PTINV are used to match exactly by country/state/county code, fire identification, location identification, and SCC. Default daily ACRESBURNED and FUEL_LOAD input are required to internally estimate heat flux for the fire-specific plume-rise computation. Fire duration information, such as BEGHOUR and ENDHOUR are used to adjust the hourly temporal profile.

Detail script setting information for fires emission modeling is described in Section 4.4.17, "Plume Rise Calculation for Fires" [167].

NOTE: All inventory pollutant names need to be listed in the master (annual) PTINV and the day-specific PTDAY files using a header record #DATA. Sample header records are shown below:

```
#ORL      FIREEMIS
#TYPE     Day-specific Point Source Inventory for FIRES
#COUNTRY US
#YEAR     2002
#DESC     FIPS,FIREID,LOCID,SCC,DATA,DATE,DATAVALUE,BEGHOUR,ENDHOUR
```

The format for the data records is shown in Table 8.21, "Format for individual day-specific data files for fires : ORL FIREEMIS" [369].

Table 8.21. Format for individual day-specific data files for fires : ORL FIREEMIS

Position	Name	Type	Description
1	FIPS	Char(5)	State and County code (required)
2	FIREID	Char	Fire idenfication code (required)
3	LOCID	Char	Location identification code (required)
4	SCC	Char(20)	SCC Flaming or Smoldering (required)
5	DATA	Char(16)	The pollutant codes, "ACRESCBURNED", or "FUEL_LOAD" (16 characters) (required)
6	DATE	Char(8)	MM/DD/YY format. Years less than 70 are treated as century 2000 (8 characters) (required)
7	DATA-VALUE	Real	If the value of "DATA" is a particular pollutant, then this value represents the daily emissions associated with the particular pollutant (required)
8	BEGHOUR	Int	Hour of the day that the fire began (0 to 23) (required)
9	ENDHOUR	Int	Hour of the day that the fire ended (0 to 23) (required)

8.2.7. PTHOUR: Point source hour-specific emissions

The PTHOUR file is the input point-source inventory file for hour-specific data and profiles. The file must be a list file (see Section 8.2.1.1, "Inventory list files" [351] and the files listed in the PTHOUR file must be in Continuous Emissions Monitoring (CEM) format when processing hourly CEM inventory data files with an format identifier (#CEM) or the format can be specified in the PTHOUR list file (#LIST CEM). In addition, these files can have a #COUNTRY entry to set the country code (see Section 8.2.1.4, "Header records" [352]. The default country code is 0, which corresponds to the United States in the default COSTCY file.

8.2.7.1. CEM hour-specific format

SMOKE uses the CEM ORIS ID and Boiler ID fields to match sources in the CEM hour-specific inventory to those in the annual inventory. Note that the CEM data hours are in standard local time (no daylight saving time adjustment). The format of the CEM hour-specific data is shown in Table 8.22, "CEM Format for individual hour-specific data files" [370].

Table 8.22. CEM Format for individual hour-specific data files

Position	Name	Type	Description
1	ORISID	Char (6)	DOE Plant Identification Code (required) (should match the same field in the `PTINV` file in IDA format)
2	BLRID	Char (6)	Boiler Identification Code (required) (should match the same field in the `PTINV` file in IDA format)
3	YYMMDD	Int	Date of data in YYMMDD format (required)
4	HOUR	Integer	Hour value from 0 to 23
5	NOXMASS	Real	Nitrogen oxide emissions (lb/hr) (required)
6	SO2MASS	Real	Sulfur dioxide emissions (lb/hr) (required)
7	NOXRATE	Real	Nitrogen oxide emissions rate (lb/MMBtu) (not used by SMOKE)
8	OPTIME	Real	Fraction of hour unit was operating (optional)
9	GLOAD	Real	Gross load (MW) (optional)
10	SLOAD	Real	Steam load (1000 lbs/hr) (optional)
11	HTINPUT	Real	Heat input (mmBtu) (required)
12	HTINPUT-MEASURE	Character(2)	Code number indicating measured or substituted, not used by SMOKE.
13	SO2MEAS-URE	Character(2)	Code number indicating measured or substituted, not used by SMOKE.
14	NOXMMEAS-URE	Character(2)	Code number indicating measured or substituted, not used by SMOKE.
15	NOXRMEAS-URE	Character(2)	Code number indicating measured or substituted, not used by SMOKE.
16	UNITFLOW	Real	Flow rate (ft3/sec) for the Boiler Unit (optional; must be present for all records or not any records not yet used by SMOKE)

The code numbers used in columns 12 through 15 have the following meanings:

- 01 = 'Measured'

- 02 = 'Calculated'

- 03 = 'Substitute'

- 04 = 'Measured and Substitute'

- 97 = 'Not Applicable'

- 98 = 'Undetermined'

- 99 = 'Unknown Code'

8.2.7.2. FF10 hour-specific format

This FF10 format uses the header described in Section 8.2.1.4, "Header records" [352].

The user specifies the data provided using this command in the header: FF10_HOURLY_POINT.
Sample header records for hourly point sources are shown below:

```
#FORMAT=FF10_HOURLY_POINT
#COUNTRY  US
#YEAR     year of inventory, e.g. 2008
#DESC     Point Hourly Source Inventory
#DESC     FF10 Point Hourly format
```

Table 8.23. FF10 Format for individual point hour-specific data files

Position	Name	Type	Description
1	COUNTRY	Char(3)	Country code: e.g., US (required)
2	FIPS	Char(5)	FIPS code for state and county (required; example 037001)
3	TRI-BAL_CODE	Char(3)	Tribal code (optional; recommended)
4	FACIL-ITY_ID	Char(15)	Facility Identification Code (required; this is the same as the Plant Identification Code in the ORL Format)
5	UNIT_ID	Char(15)	Unit Identification Code (required; this is the same as the Emission Unit ID in the NIF) (required)
6	REL_POINT_ID	Char(15)	Release Point Identification Code (required; this is the same as the Stack Identification Code in the ORL Format)
7	PRO-CESS_ID	Char(15)	Process Identification Code (recommended; this is the same as the Segment ID in the ORL Format)
8	SCC	Char(20)	SCC (required)
9	POLL	Char(16)	Pollutant Code (required)
10	OP_TYPE_CD	Char(20)	Operating type code for startup/shutdown/upset (optional; not currently used by SMOKE)
11	CALC_METH-OD	Int	CALC_METHOD (optional; not currently used by SMOKE)
12	DATE_UP-DATED	Int	Updated date in format of YYYYMMDD (optional; not currently used by SMOKE)
13	DATE	Int	Date of inventory in format of YYYYMMDD (required)
14	DAYTOT	Real	Daily Total (short tons/day) (required if `SMKINVEN_MONTH` is set to 0)
15	HRVAL0	Real	Hourly emission for hour 0 (short tons/hour) (required)
16	HRVAL1	Real	Hourly emission for hour 1 (short tons/hour) (required)
17	HRVAL2	Real	Hourly emission for hour 2 (short tons/hour) (required)
	..		..
34	HRVAL23	Real	Hourly emission for hour 23 (short tons/hour) (required)
35	COMMENT	Char	(optional)

8.2.8. `PTINV`: Point source annual or average day emissions

The `PTINV` file is the input point source inventory file for annual and average-day data. This file can be in FF10, ORL, or ORL FIRE format.

8.2.8.1. FF10 Format

The user specifies the data provided using this command in the header: FF10_POINT.

This FF10 format uses the header described in Section 8.2.1.4, "Header records" [352]. Sample header records for daily point sources are shown below:

```
#FORMAT    FF10_POINT
#COUNTRY   US
#YEAR      2005
#DESC      Point Source Inventory
#DESC      FF10 Point format
```

Table 8.24. FF10 Format for `PTINV`

Position	Name	Type	Description
1	COUNTRY	Char(3)	Country code; default to "US" (required)
2	FIPS	Char(5)	FIPS code for state and county (required)
3	TRIBAL CODE	Char(3)	Three-digit Tribal code (optional; recommended)
4	FACILITY_ID	Char(15)	Facility Identification Code (required; this is the same as Plant Identification Code in ORL Format)
5	UNIT_ID	Char(15)	Unit Identification Code (required; this is the same as the Point Identification Code in the ORL Format) (required)
6	REL_POINT_ID	Char(15)	Relative Point Identification Code (recommended; this is the same as the Stack Identification Code in the ORL Format)
7	PROCESS_ID	Char(15)	Segment Identification Code (a.k.a DOE Plant ID) (recommended; this is the same as the Process ID in the NIF)
8	AGY_FACIL-ITY_ID	Char(15)	Facility ID submitted by responsible agency (optional; not currently used by SMOKE)
9	AGY_UNIT_ID	Char	Unit ID submitted by responsible agency (optional; not currently used by SMOKE)
10	AGY_REL_POINT_ID	Char	Release Point ID submitted by responsible agency (optional; not currently used by SMOKE)
11	AGY_PRO-CESS_ID	Char	Process ID submitted by responsible agency (optional; not currently used by SMOKE)
12	SCC	Char(20)	Source Classification Code (required)
13	POLL	Char(16)	Pollutant Code (required)
14	ANN_VALUE	Real	Annual Emissions (short tons/year) (required if `SMKIN-VEN_MONTH` is set to 0)
15	ANN_PCT_RED	Real	Percentage Control Measure Reduction Efficiency (optional)
16	FACIL-ITY_NAME	Char(40)	Plant Name (required; example "Power Plant", note: use double quote around the name)
17	ERPTYPE	Char(2)	Emissions release point type (01 = fugitive, 02 = vertical stack, 03 = horizontal stack..) (optional; recommended)
18	STKHGT	Real	Stack Height (ft) (required)
19	STKDIAM	Real	Stack Diameter (ft) (required)
20	STKTEMP	Real	Stack Gas Exit Temperature (°F) (required)
21	STKFLOW	Real	Stack Gas Flow Rate (ft3/sec) (optional; automatically calculated by **Smkinven** from velocity and diameter if not given in file)
22	STKVEL	Real	Stack Gas Exit Velocity (ft/sec) (required)
23	NAICS	Char(6)	North American Industrial Classification System Code (optional)
24	LONGITUDE	Real	Longitude (decimal degrees) (required)

Position	Name	Type	Description
25	LATITUDE	Real	Latitude (decimal degrees) (required)
26	LL_DATUM	Char(3)	001=NAD27, 002=NAD83, 003=WGS84 (optional; not currently used by SMOKE)
27	HOR-IZ_COLL_MTHD	Char	Flag indicating how NEI defaulted locations (optional; not currently used by SMOKE)
28	DESIGN_CA-PACITY	Real	Unit Design Capacity (optional; not currently used by SMOKE)
29	DESIGN_CA-PA-CITY_UNITS	Char	Unit Design Capacity Units (optional; not currently used by SMOKE)
30	REG_CODES	Char	Regulatory Code (optional; not currently used by SMOKE)
31	FAC_SOURCE_TYPE	Real	Facility source type (optional; not currently used by SMOKE)
32	UNIT_TYPE_CODE	Real	Unit type code (optional; not currently used by SMOKE)
33	CON-TROL_IDS	Char	Control Measure Code (optional; not currently used by SMOKE)
34	CON-TROL_MEAS-URES	Char	Control Measure Values (optional; not currently used by SMOKE)
35	CUR-RENT_COST	Real	Control Cost (optional; not currently used by SMOKE)
36	CUMULAT-IVE_COST	Real	Cumulative Cost (optional; not currently used by SMOKE)
37	PROJEC-TION_FACTOR	Real	Projection factor (optional; not currently used by SMOKE)
38	SUBMIT-TER_FAC_ID	Char(15)	Primary Facility ID (optional; not currently used by SMOKE)
39	CALC_METH-OD	Int	Emission Calculation Method Code (optional; not currently used by SMOKE)
40	DATA_SET_ID	Int	Data Set Identifier (optional; not currently used by SMOKE)
41	FACIL_CAT-EGORY_CODE	Char	Facility Site Identifier (optional; not currently used by SMOKE)
42	ORIS_FACIL-ITY_CODE	Char	DOE Plant ID (generally recommended, and required if matching to hour-specific CEM data)
43	ORIS_BOIL-ER_ID	Char	Boiler Identification Code (recommended)
44	IPM_YN	Char	Y if point source belongs to IPM Sector (optional; not currently used by SMOKE)
45	CALC_YEAR	Int	Year used to represent 2002 emissions (optional; not currently used by SMOKE)
46	DATE_UP-DATED	Int	Last updated date of emissions table (optional; not currently used by SMOKE)

Position	Name	Type	Description
47	FUG_HEIGHT	Real	Fugitive Height(optional; recommended)
48	FUG_WIDTH_YDM	Real	Fugitive Width(optional; recommended)
49	FUG_LENGTH_XDM	Real	Fugitive Length(optional; recommended)
50	FUG_ANGLE	Real	Fugitive Angle(optional; recommended)
51	ZIPCODE	Int	Location Address Postal Code (optional; not currently used by SMOKE)
52	ANNU-AL_AVG_HOURS_PER_YEAR	Int	Annual Average Hours of Operation per Year (optional; not currently used by SMOKE)
53	JAN_VALUE	Real	Filled in only if data were provided for January in unit of short tons/month (required if `SMKINVEN_MONTH` is set to 1)
...	...	...	...
64	DEC_VALUE	Real	Filled in only if data were provided for December in unit of short tons/month (required if `SMKINVEN_MONTH` is set to 12)
65	JAN_PCTRED	Real	Month-specific percent reduction. Filled in only if data were provided for January in unit of % (optional; not currently used by SMOKE)
...	...	...	...
76	DEC_PCTRED	Real	Month-specific percent reduction. Filled in only if data were provided for December in unit of % (optional; not currently used by SMOKE)
77	COMMENT	Char	Optional; not currently used by SMOKE

8.2.8.2. ORL Format

The ORL format is list-directed and uses the header described in Section 8.2.1.4, "Header records" [352]. Sample header records are shown below:

```
#ORL POINT
#TYPE      Point Source Toxics Inventory
#COUNTRY   US
#YEAR      1999
#DESC      Alabama 1999
```

The format for the data records is shown in Table 8.25, "ORL Format for PTINV" [376].

Table 8.25. ORL Format for `PTINV`

Position	Name	Type	Description
1	FIPS	Char(5)	FIPS code for state and county (required)
2	PLANTID	Char(15)	Plant Identification Code (required; this is the same as the State Facility Identifier in the NIF)
3	POINTID	Char(15)	Point Identification Code (required; this is the same as the Emission Unit ID in the NIF)
4	STACKID	Char(15)	Stack Identification Code (recommended; this is the same as the Emissions Release Point ID in the NIF)
5	SEGMENT	Char(15)	DOE Plant ID (recommended; this is the same as the Process ID in the NIF)
6	PLANT	Char(40)	Plant Name (recommended; example "Power Plant", note: use double quote around the name)
7	SCC	Char(20)	Source Classification Code (required)
8	ERPTYPE	Char(2)	Emissions release point type; indicates type of stack (optional; recommended) • 01 = fugitive • 02 = vertical stack • 03 = horizontal stack • 04 = goose neck • 05 = vertical with rain cap • 06 = downward-facing vent
9	SRCTYPE	Char(2)	Source type; used by SMOKE in determining applicable MACT-based controls (required) • 01 = major • 02 = Section 12 area source • 03 = nonroad • 04 = onroad
10	STKHGT	Real	Stack Height (ft) (required)
11	STKDIAM	Real	Stack Diameter (ft) (required)
12	STKTEMP	Real	Stack Gas Exit Temperature (°F) (required)
13	STKFLOW	Real	Stack Gas Flow Rate (ft^3/sec) (optional; automatically calculated by **Smkinven** from velocity and diameter if not given in file)
14	STKVEL	Real	Stack Gas Exit Velocity (ft/sec) (required)
15	SIC	Char(20)	Standard Industrial Classification Code (recommended)
16	MACT	Char(6)	Maximum Available Control Technology Code (optional)
17	NAICS	Char(6)	North American Industrial Classification System Code (optional)

Position	Name	Type	Description
18	CTYPE	Char(1)	Coordinate system type (required) • U = Universal Transverse Mercator • L = Latitude/longitude
19	XLOC	Real	X location (required) • If CTYPE = U, Easting value (meters) • If CTYPE = L, Longitude (decimal degrees)
20	YLOC	Real	Y location (required) • If CTYPE = U, Northing value (meters) • If CTYPE = L, Latitude (decimal degrees)
21	UTMZ	Int	UTM zone (required if CTYPE = U)
22	CAS	Char(16)	Pollutant CAS number or other code (required; this is called the pollutant code in the NIF)
23	ANN_EMIS	Real	Annual Emissions (short tons/year) (required)
24	AVD_EMIS	Real	Average-day Emissions (short tons/average day) (optional)
25	CEFF	Real	Control Efficiency percentage (give value of 0-100) (recommended, if left blank, SMOKE default is 0)
26	REFF	Real	Rule Effectiveness percentage (give value of 0-100) (recommended, if left blank, SMOKE default is 100)
27	CPRI	Int	Primary Control Equipment Code (not used by SMOKE)
28	CSEC	Int	Secondary Control Equipment Code (not used by SMOKE)
29	NEI_UNIQUE_ID	Char	Unique ID that ties together HAP and CAP emissions within a common facility ID and also ties together emissions obtained from muptiple data sources which may have different State facility identifiers but really belong to a single facility (optional; not currently used by SMOKE)
30	ORIS_FACIL-ITY_CODE	Char	DOE Plant ID (generally recommended, and required if matching to hour-specific CEM data)
31	ORIS_BOIL-ER_ID	Char	Boiler Identification Code (recommended)
32	IPM_YN	Char	Y or N character flag indicating whether the point source belongs in the IPM sector or not (optional; not currently used by SMOKE)
33	DATA_SOURCE	Char	Flag indicating the source of the data (e.g., state submitted or EPA-computed) (optional; not currently used by SMOKE)
34	STACK_DE-FAULT_FLAG	Char	Flag indicating how NEI defaulted the stack parameters (optional; not currently used by SMOKE)

Position	Name	Type	Description
35	LOCA-TION_DE-FAULT_FLAG	Char	Flag indicating how NEI defaulted locations (optional; not currently used by SMOKE)
36	YEAR	Char	Year of the emissions (optional; not currently used by SMOKE)
37	TRI-BAL_CODE	Char(3)	Indicates the particular tribe that submitted the data (optional; recommended)
38	HORIZONT-AL_AREA_FU-GITIVE	Real	Related to AXLEN and AYLEN which are optional parameters for ISCST3/AERMOD (optional; not currently used by SMOKE)
39	RE-LEASE_HEIGHT_FU-GITIVE	Real	Related to optional parameters for ISCST3/AERMOD (optional; not currently used by SMOKE)
40	ZIPCODE	Real(14)	Postal zip code (optional; not currently used by SMOKE)
41	NAICS_FLAG	Real	NAICS flag (optional; not currently used by SMOKE)
42	SIC_FLAG	Real	SIC flag (optional; not currently used by SMOKE)
43	MACT_FLAG	Char(15)	MACT flag indicating the relationship of the MACT rule and the particular source (optional; not currently used by SMOKE)
44	PRO-CESS_MACT_COM-PLI-ANCE_STATUS	Char	Indicate the relationship of the MACT rule and the particular source (optional; not currently used by SMOKE)
45	IPM_FACIL-ITY	Char	Integrated Planning Model (IPM) facility (optional; not currently used by SMOKE)
46	IPM_UNIT	Char	IPM unit (optional; not currently used by SMOKE)
47	BART_SOURCE	Char	Best Available Retrofit Technology (BART) source (optional; not currently used by SMOKE)
48	BART_UNIT	Char	BART unit (optional; not currently used by SMOKE)
49	CON-TROL_STATUS	Char	Control status (optional; not currently used by SMOKE)
50	START_DATE	Char(8)	Indicates originally-reported start date prior to annualizing emissions, which may be used for building temporal profiles for specific sources (optional; not currently used by SMOKE)
51	END_DATE	Char(8)	Indicates originally-reported end date prior to annualizing emissions, which may be used for building temporal profiles for specific sources (optional; not currently used by SMOKE)
52	WINTER_THROUGH-PUT_PCT	Real	Percentage throughout in winter (optional; not currently used by SMOKE)
53	SPRING_THROUGH-PUT_PCT	Real	Percentage throughout in spring (optional; not currently used by SMOKE)

Position	Name	Type	Description
54	SUM-MER_THROUGH-PUT_PCT	Real	Percentage throughout in summer(optional; not currently used by SMOKE)
55	FALL_THROUGH-PUT_PCT	Real	Percentage throughout in fall (optional; not currently used by SMOKE)
56	ANNU-AL_AVG_DAYS_PER_WEEK	Real	Annual average days of operation per week (optional; not currently used by SMOKE)
57	ANNU-AL_AVG_WEEKS_PER_YEAR	Real	Annual average weeks of operation per year (optional; not currently used by SMOKE)
58	ANNU-AL_AVG_HOURS_PER_DAY	Real	Annual average hours of operation per day (optional; not currently used by SMOKE)
59	ANNU-AL_AVG_HOURS_PER_YEAR	Real	Annual average hours of operation per year (optional; not currently used by SMOKE)
60	PERI-OD_DAYS_PER_WEEK	Real	Days per week operating during the period (optional; not currently used by SMOKE)
61	PERI-OD_WEEKS_PER_PERI-OD	Real	Weeks per period operating during the period (optional; not currently used by SMOKE)
62	PERI-OD_HOURS_PER_DAY	Real	Hours of operation per day during the period (optional; not currently used by SMOKE)
63	PERI-OD_HOURS_PER_PERI-OD	Real	Hours of operation per period during the period (optional; not currently used by SMOKE)
64	DESIGN_CA-PACITY	Real	Indicate the design capacity for the source (optional; not currently used by SMOKE)
65	DESIGN_CA-PA-CITY_UNIT_NU-MERATOR	Real	Indicate the numerator for the units of the design capacity (optional; not currently used by SMOKE)
66	DESIGN_CA-PA-CITY_UNIT_DE-NOMINATOR	Real	Indicate the denumerator for the units of the design capacity (optional; not currently used by SMOKE)
67	CON-TROL_MEAS-URES	Char	An & delimited list of control measure abbreviations (optional; not currently used by SMOKE)
68	PCT_REDUC-TION	Char	An & delimited list of percent reductions for the applied control measures (optional; not currently used by SMOKE)
69	CUR-RENT_COST	Char	The cost of the most recently applied control strategy for this source (optional; not currently used by SMOKE)
70	CUMULAT-IVE_COST	Char	The cumulative cost for all applied control measures (optional; not currently used by SMOKE)

8.2.8.3. ORL FIRE Format

The ORL inventory file for fires (#ORL FIRE) is one of the required input files to provide wildfire and prescribed fire data that are being treated as point sources to SMOKE. This file includes the fire characteristics, such as county/state/county code, fire identification, fire name, location identification, location coordinates, SCC (flaming or smoldering), material burned, vegetation types and heat content. Some of characteristics that are common to this master (annual) inventory file PTINV and the day-specific inventory file PTDAY shown in Section 8.2.6.2, "ORL FIRE Emission Format" [368] need to match exactly by county/state/county code, fire ID, location ID, location coordinates, and SCC.

Detail script setting information for fires emission modeling is described in Section 4.4.17, "Plume Rise Calculation for Fires" [167].

NOTE: Although inventory pollutants are not listed, user needs to list all inventory pollutant names available from the day-specific inventory file PTDAY using a header record #DATA. Sample header records are shown below:

```
#ORL     FIRE
#TYPE    Point Source Inventory for FIRES
#COUNTRY US
#YEAR    2002
#DATA    FUEL_LOAD ACRESBURNED CO NOX SO2 PM10 PM2_5 VOC 16990 10883
#DESC FIPS,FIREID,LOCID,SCC,NAME,LAT,LON,NFDRSCODE,MATBURNED,HEATCONTENT
```

The format for the data records is shown in Table 8.26, "ORL FIRE Format for PTINV" [380].

Table 8.26. ORL FIRE Format for PTINV

Position	Name	Type	Description
1	FIPS	Char(5)	FIPS code for state and county (required)
2	FIREID	Char(15)	Fire Identification Code (required)
3	LOCID	Char(15)	Location Identification Code (required)
4	SCC	Char(20)	Source Classification Code (required)
5	FIRENAME	Char(40)	Fire name (recommended)
6	LAT	Real	Latitude associated with FIREID/LOCID
7	LON	Real	Longitude associated with FIREID/LOCID
8	NFDRSCODE	Char(2)	Code for the Vegetation type (recommended)
9	MATBURNED	Int	Code for the material burned
10	DATAVALUE	Real	Value of heat content

8.3. Temporal Allocation Files

The directories where the temporal allocation files should be stored are listed in Table 8.2, "Temporal Allocation Input File Locations" [347]. The table also lists the programs that use each input file.

8.3.1. `[A|M|P]TPRO`: Temporal profiles files

The Temporal profiles files contain the monthly, weekly, diurnal and/or hourly temporal profiles used to convert the annual emissions stored in the inventory to hourly emissions. The files can contain the following types of records:

- `[A|M|P]TPRO_MONTHLY` : Year to month temporal weighting factors

- `[A|M|P]TPRO_DAILY` : Year to day temporal weighting factors

- `[A|M|P]TPRO_WEEKLY` : Month to Week temporal weighting factors

- `[A|M|P]TPRO_HOURLY` : Day to hour temporal weighting factors

- `[A|M|P]TPRO_HOURLY_NCF` : Hourly temporal weighting factors

This optional date-specific hourly temporal profiles will take precedence over the default monthly, weekly and hourly profiles. Also users must to use FIPS and SCC cross-reference hiearchy to use these temporal profiles.

8.3.1.1. [A|M|P]TPRO_MONTHLY: Year-to-Monthk Temporal Profiles

Table 8.27. Format for `ATPRO_MONTHLY`, `MTPRO_MONTHLY`, and `PTPRO_MONTHLY`

Position	Type	Description
0	Chars(15)	Monthly Temporal Profile ID (15 characters)
1	Real	Temporal factor for January
2	Real	Temporal factor for February
3	Real	Temporal factor for March
...	Real	.....
10	Real	Temporal factor for October
11	Real	Temporal factor for November
12	Real	Temporal factor for December
13	Chars	Comments (optional: must be double (quoted)

8.3.1.2. [A|M|P]TPRO_DAILY: Month-to-day Temporal Profiles

Table 8.28. Format for `ATPRO_DAILY`, `MTPRO_DAILY`, and `PTPRO_DAILY`

Position	Type	Description
0	Chars(15)	Month-to-day Temporal Profile ID (15 characters)
1	Integer	Calendar Month [ex: 1, 2,,,,12]
2	Real	Temporal factor for Day 1
3	Real	Temporal factor for Day 2
...	Real	.....
30	Real	Temporal factor for Day 29
31	Real	Temporal factor for Day 30
32	Real	Temporal factor for Day 31
33	Chars	Comments (optional: must be double (quoted)

8.3.1.3. [A|M|P]TPRO_WEEKLY: Month-to-Week Temporal Profiles

Table 8.29. Format for `ATPRO_WEEKLY`, `MTPRO_WEEKLY`, and `PTPRO_WEEKLY`

Position	Type	Description
0	Chars(15)	Weekly Temporal Profile ID (15 characters)
1	Real	Temporal factor for Monday
2	Real	Temporal factor for Tuesday
3	Real	Temporal factor for Wednesday
4	Real	Temporal factor for Thursday
5	Real	Temporal factor for Friday
6	Real	Temporal factor for Saturday
7	Real	Temporal factor for Sunday
8	Chars	Comments (optional: must be double (quoted)

8.3.1.4. [A|M|P]TPRO_HOURLY: Day-to-Hour Temporal Profiles

Table 8.30. Format for `ATPRO_HOURLY`, `MTPRO_HOURLY`, and `PTPRO_HOURLY`

Position	Type	Description
0	Chars(15)	Hourly Temporal Profile ID (15 characters)
1	Real	Temporal factor for Hour beginning 0000
2	Real	Temporal factor for Hour beginning 0100
3	Real	Temporal factor for Hour beginning 0200
...	Real	.....
22	Real	Temporal factor for] Hour beginning 2100
23	Real	Temporal factor for Hour beginning 2200
24	Real	Temporal factor for Hour beginning 2300
25	Chars	Comments (optional: must be double (quoted)

8.3.1.5. [A|M|P]TPRO_HOURLY_NCF: Hourly Temporal Profiles

This NetCDF hourly temporal profiles input file contains annual total (`ANNTOT`), monthly total (`MONTOT`), daily total (`DAYTOT`) and hourly total (`HRLTOT`) to compute year-to-hour, month-to-hour, day-to-hour profiles to calculate hourly total depending on types of inventory data (annual, monthly or daily inventory). If inventory is annual total, year-to-hour profiles will be used to convert annual total to hourly total emissions. If inventory is monthly total, month-to-hour profiles will be used to compute hourly total from monthly total inventory.

Table 8.31. Format for `ATPRO_HOURLY_NCF`, `MTPRO_HOURLY_NCF`, and `PTPRO_HOURLY_NCF`

Variable Name	Type	Description
Profile ID	Integer	Hourly Temporal Profile ID (a.k.a. FIPS code)
ANNTOT	Real	Annual Total by profile ID
MONTOT	Real	Monthl Total by profile ID
DAYTOT	Real	Daily Total by profile ID
HRLTOT	Real	Hourly Total by profile ID

8.3.2. [A|M|P]TREF: Temporal cross-reference file

The `ATREF`, `MTREF`, and `PTREF` files contain the area-, mobile-, and point-source cross-reference records for assigning the temporal profiles to the sources. If needed, the same file can be used for area-, mobile-, or point-source categories, but more fields are used for mobile than for area, and more fields are used for point than for mobile. Section 6.16, "**Temporal**" [291] provides a list of the matching hierarchies that SMOKE uses when applying the file to an inventory. If a field is blank, it must be set to -9.

Between weekly and daily temporal profiles, daily Profiles will take precedence weekly profiles. Within hourly diurnal profiles, the processing occurs as follows (1) day-to-hour hourly profiles (i.e., Monday,,,,,Sunday), (2) Weekday/Weekend hourly profiles, and (3) All days (Monday-Sunday) hourly profiles. However, date-specific houlry temporal profiles from [A|M|P]TPRO_HOURLY_NCF input file will take precedence any types of temporal profiles. This date-specific hourly temporal profiles NetCDF input file contains annual total (`ANNTOT`), monthly total (`MONTOT`), daily total (`DAYTOT`) and hourly total

(HRLTOT) to compute year-to-hour, month-to-hour, day-to-hour profiles to calculate hourly total depending on types of inventory data (annual, monthly or daily inventory). If inventory is annual total, year-to-hour profiles will be used to convert annual total to hourly total emissions. If inventory is monthly total, month-to-hour profiles will be used to compute hourly total from monthly total inventory.

Within the **Temporal** program, the processing occurs as follows:

- Temporal profiles are assigned for all sources.

- Monthly (year-to-month) temporal profiles can be applied.

- Daily (month-to-day) temporal profiles can be applied.

- Weekly (month-to-weekly) temporal profiles can be applied.

- Day-of-week specific hourly diurnal temporal profiles (Monday,,,Sunday, Weekday, Weekend, or All days) are applied.

- Date-specific hourly diurnal temporal profiles (year-to-hour, month-to-hour or day-to-hour) can be applied.

- Day-specific emissions or activities are read in and used to overwrite daily emissions for the appropriate sources

- Hour-specific emissions or activities are read in and used to overwrite hourly emissions for the appropriate sources.

NOTE: Date-specific houlry diurnal temporal profiles require FIPS and SCC specific entries in [A|M|P]TREF input file.

The pollutant field allows the user to assign a different temporal profile for each pollutant associated with a given source. This is an optional approach; if you do not want to use pollutant as a characteristic in assigning temporal profiles, you can put a "-9" in the pollutant field. In that case, all pollutants for a given source will be assigned that temporal profile.

The country/state/county code is defined as follows:

000000 = applies to all counties, states, and counties
Y00000 = applies to all states and counties in country Y
YSS000 = applies to all counties in state SS and country Y
YSSCCC = applies to county CCC in state SS and country Y

Table 8.32. Format for `[A|M|P]TREF`

Line	Position	Description
1+	A	20-digit SCC, or 10-digit SCC or zeros filling the remaining digits (Character) (optional; put zero for entry that is not SCC-specific)
	B	Country/state/county code (6-digit Character in format YSSCCC) (optional)
	C	Facility ID (for ORL, this is Plant ID) (Character(15)) (optional; point source only, leave blank for not facility-specific)
	D	Unit ID (for ORL, this is point ID) (Character(15)) (optional: point source only)
	E	Release Point (for ORL, this is stack ID) (Character(15)) (optional: point source only)
	F	Process ID (for ORL, this is segment ID) (Character(15)) (optional: point source only)
	G	Pollutant name (Character) (optional; put zero if not pollutant-dependent)
	H	"**HOURLY**": Date-specific hourly diurnal temporal profile from `[A|M|P]TPRO_HOURLY_NCF`. "**MONTHLY**": Year to month temporal profile from `[A|M|P]TPRO_MONTHLY`. "**DAILY**": Year to day temporal profile from `[A|M|P]TPRO_DAILY`. "**WEEKLY**": Month to week temporal profile from `[A|M|P]TPRO_WEEKLY`. "**MONDAY**": Monday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**TUESDAY**": Tuesday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**WEDNESDAY**": Wendesday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**THURSDAY**": Thursday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**FRIDAY**": Friday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**SATURDAY**": Saturday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**SUNDAY**": Sunday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**WEEKDAY**": Weekday diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**WEEKEND**": Weekend diurnal temporal profile from `[A|M|P]TPRO_HOURLY`. "**ALLDAY**": All days diurnal temporal profile from `[A|M|P]TPRO_HOURLY`.
	I	Temporal Profile ID (Characters) (required)
	J	Comments (Characters) (optional; must be double quoted)

8.3.3. `HOLIDAYS`: Holidays date list

SMOKE can process holidays as specific days of the week in order to better model activity patterns. The `HOLIDAYS` file is used to indicate which days should receive special treatment and how the days should be treated. The format of this file is shown in Table 8.33, "Format for `HOLIDAYS`" [386].

Table 8.33. Format for `HOLIDAYS`

Line	Position	Description
1+	A	Five-digit FIPS code, only 00000 is currently supported (required)
	B	Month of holiday (1 - 12) (required)
	C	Day of holiday (1 - 31) (required)
	D	Year of holiday (4 digits) (required)
	E	Day of the week to treat holiday as, valid values are Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday (required)

8.3.4. `PROCDATES`: Procdates date list

SMOKE can process specific sequential or non-seqential days of the year. The `PROCDATES` file is used to indicate which days should be processed temporally. The format of this file is shown in Table 8.34, "Format for `PROCDATES`" [386].

Table 8.34. Format for `PROCDATES`

Line	Position	Description
1+	A	Seven digit Gregorian date (YYYYMMDD) (required)
	B	Six digit Start time (HHMMSS) (required)
	C	Six digit Run length (HHMMSS) (required)

8.4. Spatial Allocation Files

All spatial allocation (gridding) inputs need to be consistent. Each file has a header to indicate the grid projection, location, dimensions, and resolution; SMOKE uses these headers to ensure consistency across all inputs. SMOKE also requires that any meteorology data inputs be consistent with these gridded data. Most programs in SMOKE are capable of processing a modeling grid that is a subgrid of the input data (i.e. spatial surrogates, meteorology data).

Several options are available for creating the spatial surrogate and biogenic land use inputs.

- The MIMS Spatial Allocator can create surrogates and prepare BELD3 inputs files for any modeling grid.

- Several national surrogate data sets are available online at EPA's Clearinghouse for Inventories and Emissions Factors [http://www.epa.gov/ttn/chief/emch/spatial/]. These data sets are at 36-, 12-, and 4-km resolutions.

- The gridding surrogates and land use data can be prepared using a GIS program.

The directories where the spatial allocation files should be stored are listed in Table 8.3, "Spatial Allocation Input File Locations" [348]. The table also lists the programs that use each input file.

8.4.1. `SRGDESC`: Surrogate Designation, Description and Surrogate profile filename.

The Surrogate Description file is used to designate the regional location of the surrogate, the surrogate code number, a brief description of the code number and the Area and Mobile spatial surrogate file in which surrogate profile can be found. This file is located in `SRGPRO_PATH`. See Figure 6.4, "**Grdmat** input and output files" [230]. The format of the `SRGDESC` file is shown in Table 8.35, "Format for `SRG-DESC`" [387].

Table 8.35. Format for `SRGDESC`

Line	Position	Description
1	A	#GRID
	B	Grid name
	C	X origin in units of the projection
	D	Y origin in units of the projection
	E	X direction cell length in units of the projection
	F	Y direction cell length in units of the projection
	G	Number of columns
	H	Number of rows
	I	Number of boundary cells
	J	Projection types: • Latitude-Longitude: "LAT-LON" or "LATGRD3" • Lambert Conformal: "LAMBERT" or "LAMGRD3" • Polar Stereographic: "POLAR" or "POLGRD3" • UTM: "UTM" or "UTMGRD3"
	K	Projection units
	L	Projection alpha value
	M	Projection beta value
	N	Projection gamma value
	O	X-direction projection center in units of the projection
	P	Y-direction projection center in units of the projection
2+	A	Region of surrogate i.e. USA, MEX, or CAN. (ASCII) (not used by SMOKE)
	B	Surrogate code number (Integer)(required)
	C	Surrogate Description (ASCII 80 characters in quotes) (required)
	D	Surrogate file name (ASCII 80 characters) (required) located in `SRGPRO_PATH`

8.4.2. Area and Mobile Spatial Surrogates Files

The spatial surrogates files contain the spatial allocation factors for all area sources and non-link mobile sources. One file may be used for both area and mobile sources, however, multiple surrogate files may also be used to futher separate and refine the detail of the spatial allocation. The location of these files can

be found in Table 8.3, "Spatial Allocation Input File Locations" [348] The format for these files is shown in Table 8.36, "Format for Area and Mobile spatial surrogate files." [388].

Table 8.36. Format for Area and Mobile spatial surrogate files.

Line	Position	Description
1	A	#GRID
	B	Grid name
	C	X origin in units of the projection
	D	Y origin in units of the projection
	E	X direction cell length in units of the projection
	F	Y direction cell length in units of the projection
	G	Number of columns
	H	Number of rows
	I	Number of boundary cells
	J	Projection types: • Latitude-Longitude: "LAT-LON" or "LATGRD3" • Lambert Conformal: "LAMBERT" or "LAMGRD3" • Polar Stereographic: "POLAR" or "POLGRD3" • UTM: "UTM" or "UTMGRD3"
	K	Projection units
	L	Projection alpha value
	M	Projection beta value
	N	Projection gamma value
	O	X-direction projection center in units of the projection
	P	Y-direction projection center in units of the projection
2+	A	Spatial surrogates code (area) or county feature/roadway type (mobile) (Integer) (required)
	B	Country/state/county code (6-digit Character YSSCCC) (required)
	C	Grid column number (Integer) (required)
	D	Grid row number (Integer) (required)
	E	Spatial surrogate ratio (area) or fraction of county feature in cell (mobile) (Real) (required)

8.4.3. AGREF and MGREF: Gridding cross-reference files for area and mobile sources

The gridding cross-reference files are used to match the emissions sources with the spatial surrogates in the surrogates file. The way that of the gridding cross-reference files are applied in SMOKE ensures that the most-specific matches override the less-specific ones. A description of how the gridding cross-reference is applied to the inventory is provided in Section 6.4, "**Grdmat**" [229]. The format of the AGREF and MGREF files is shown in Table 8.37, "Format for AGREF and MGREF" [389].

Table 8.37. Format for `AGREF` and `MGREF`

Line	Position	Description
1+	A	Country/state/county code (6-digit Character YSSCCC) (optional): • leave blank or put zero if not country/state-dependent • set to YSS000 for country/state-dependent • set to YSSCCC for county-dependent
	B	20-digit SCC (Character) (optional, set to zero if not SCC-dependent)
	C	Spatial surrogate code (Integer) (required)

8.4.4. `BELD3_A`, `BELD3_B`, `BELD3_TOT`: BELD3 land use data files

`BELD3_A`, `BELD3_B`, and `BELD3_TOT` files are I/O API netCDF files that contain gridded BELD3 data. The BELD3 data comprise 230 land use types. Because the I/O API limit on the number of variables in one netCDF file is 120, the gridded BELD3 data is broken up into two files (`BELD_A` and `BELD_B`). The `BELD_TOT` file contains additional information on the BELD3 data for the same grid as the `BELD3_A` and `BELD3_B` files. The `BELD3_TOT` file contains the following variable and description:

• FOREST: Percentage of forest area in each grid cell

8.4.5. `BGPRO`: Biogenic spatial surrogate file

The `BGPRO` file must be provided to SMOKE when using county land use data, in order to provide a means of converting county land use to gridded land use. The format for the `BGPRO` file is the same as for the Area and Mobile spatial surrogate files (see Section 8.4.2, "Area and Mobile Spatial Surrogates Files" [387]).

8.4.6. `GRIDDESC`: Grids and projection information

The `GRIDDESC` file is used to provide grid and projection information to SMOKE and can contain data for many different projections and many different grids. The format of the `GRIDDESC` file is available in the I/O API documentation at http://www.baronams.com/products/ioapi/GRIDDESC.html.

8.5. Chemical speciation

The directories where the chemical speciation files should be stored are listed in Table 8.4, "Chemical Speciation Input File Locations" [348]. The table also lists the programs that use each input file.

8.5.1. `GSCNV`: Pollutant-to-pollutant conversion file

The pollutant-to-pollutant conversion factors file is required when there is a mismatch between the pollutant in the inventory and the pollutant for which the speciation profiles have been developed. The typical example is when the inventory has VOC (or ROG), but the speciation profiles are developed for TOG. SMOKE permits such a factor to be applied to any pollutant. The same format is used by all source categories.

These factors can be applied to the inventory using one of two assignment approaches: either by FIPS/SCC or by Speciation Profile ID. The same approach must be used for all sources in a single run of Spcmat. Historically in SMOKE, the FIPS/SCC approach has been used; however, starting with SMOKE version

2.3, users may instead apply these factors by Speciation Profile ID. For assignment of factors using the state/county FIPS and SCC codes, the available combinations and hierarchies for applying the factors are the same as for the speciation cross-reference (see Section 6.15, "**Spcmat**" [286]), with the exception of the plant ID and other more specific source characteristics for point sources. For assignment of the factors using Speciation Profile ID, there is no assignment hierarchy - whatever profile was assigned to a source by the GSREF file will be used to choose the factor to apply from the GSCNV file, for the appropriate pollutants. To assign these factors by Speciation Profile ID, a special header line is needed, as described below. SMOKE version 2.3 is backward compatible with previous GSCNV files used with earlier versions.

The GSCNV file is not needed if there is no mismatch between the inventory and the pollutants in the speciation profiles file. The **Spcmat** program will input and use this file only when the POLLUTANT_CON-VERSION environment variable is set to Y.

If emissions types (emission process/pollutant combinations) are being used, as with mobile source processing using the **Movesmrg** program (see Section 2.8.4.9, " MOVES Emission Processes by Emissoin Rate Tables" [43]), the emission type must be listed instead of the pollutant. For example, if exhaust running VOC emissions were being converted to exhaust TOG, then EXR__VOC and EXR__TOG would need to appear in their respective columns on the header line (line 1 in the "old" format below or in the first and second columns in the "new" format). It would not be correct to put only VOC and TOG. All emission types must be listed, or the conversion factor that will be applied is 1. This approach permits conversion factors that are specific to each emission process (e.g., different conversions for EXR and EXS).

The format of the GSCNV file for Speciation Profile ID assignments is shown in Table 8.38, "Format for GSCNV (using Speciation Profile ID assignments)" [390]. The format is consistent with the format created by the Emission Modeling Framework Speciation Tool.

Table 8.38. Format for GSCNV (using Speciation Profile ID assignments)

Line	Position	Description
1	1-11	#BY PROFILE (required)
2	A	Name of pollutant or emission type converting from (16-character) (required)
	B	Name of pollutant or emission type converting to (16-character) (required)
	C	Speciation profile code (5-character) (required)
	D	Volatile conversion factor (Real) (required)
Repeat line 2 as needed for multiple pollutants and/or emission types		

8.5.2. GSPRO: Speciation profile file

The speciation profile file, GSPRO, contains the factors that are used to separate aggregated inventory pollutant emissions totals into emissions of model species required by the AQM. In traditional ozone air quality modeling, this file contained split factors only for VOC. In SMOKE, all inventory pollutants need to be represented in this file, even if it is only to indicate that a pollutant is not split into any model species (e.g., CO emissions would have a "split factor" of 1.0). The format of the GSPRO file is shown in Table 8.39, "Format for GSPRO" [391].

For support of combining criteria and toxic VOC using the NONHAPVOC calculation feature of SMOKE, the GSPRO file (starting in version 2.3 of SMOKE) includes a special header that defines the meaning of NONHAPVOC. This is the new #NHAP header, described below. This header keyword will be treated as a comment line by previous versions of SMOKE. The header lines define NONHAPVOC (and/or NON-HAPTOG) by listing all of the VOC compounds that were assumed to be removed from VOC (or TOG) when the speciation profile factors for NONHAPVOC (or NONHAPTOG) were calculated. When using

this file, Spcmat compares this list with the list from the Inventory Table ("VOC or TOG Component" field = V or T and "Explicit in Mechanism = "N"). This new header is included in this file so that the file is self-describing with respect to the definition of NONHAPVOC and NONHAPTOG.

The format supports header lines (using a # symbol in position 1 of any line) in any part of the file.

Table 8.39. Format for `GSPRO`

Line	Position	Description (all fields are required)
1+	A	#NHAP (optional header lines to be included only for NONHAPVOC processing approach in SMOKE. Skip to Line 2 for all other cases)
	B	Pollutant name (either NONHAPVOC or NONHAPTOG)
	C	Toxics name of compound(s) subtracted from VOC or TOG to calculate NONHAPVOC or NONHAPTOG (must match "Inventory Data Name" column from Inventory Table (`INVTABLE`)
2+	A	Speciation profile number (Character)
	B	Pollutant ID (name) (Character)
	C	Species ID (name) (Character)
	D	Split factor (Real)
	E	Divisor (Real)
	F	Mass Fraction (Real)

The columns are used by SMOKE as follows:

- The speciation profile number is matched with the number in the GSREF file. This "number" is treated as a character string, so leading zeros must be the same in both the GSPRO and GSREF files.

- The pollutant ID must match the pollutant names in the SMOKE inventory file (`AREA`, `MOBL`, or `PNTS`). These names are set by the #POLID or #DATA headers in the **Smkinven** input files.

- The species IDs are the names of the model species needed by the AQM for which SMOKE is being used. These depend on the requirements of that model. The example SMOKE inputs will not be correct for all AQMs.

- The split factor is the numerator in the factor computed for the mole-based speciation matrix.

- The divisor is the denominator in the factor computed for the mole-based speciation matrix.

- The mass fraction is the value put in the mass-based speciation matrix. The ratio of the mass fraction and the split factor is the average molecular weight for the species and profile, if the divisor is 1.

8.5.3. `GSPRO_COMBO`: **Combination speciation profiles file**

When the special keyword "COMBO" is used as a speciation profile code in the Section 8.5.4, "GSREF: Speciation cross-reference file" [392], all sources matching the cross-reference entry should use a combination speciation profile. The combination speciation profiles file (GSPRO_COMBO) contains a combination of multiple speciation profiles that combine multiple speciation profiles with a weight for each profile to support a mixture of fuel profiles to specific counties, when users have information to provide about market penetration of each fuel and have speciation profiles for each of these fuels available. The format of the GSPRO_COMBO file is shown in Table 8.40, "Format for GSPRO_COMBO" [392].

Table 8.40. Format for `GSPRO_COMBO`

Position	Name	Description
A	POLID	Pollutant ID (Char 16) (required, should be MODE__POLLUTANT for entries that are mode-pollutant combinations).
B	FIPS	FIPS code for country, state and county (Char(5)) (required)
C	PERIOD	Period number(e.g., month number)(integer) (required). Need to match SPCMAT_PERIOD settting in script for data that are not period specific, use 1 as default.
D	NPROF	Number of profiles in the combination (integer) (required). Values less than or equal to 0 will cause record to be skipped. The maximum number of profiles allowed is 10.
E	PROF[1]	First speciation profile code (Char 5) (required). Must match profile code in GSPRO file for pollutant ID.
F	FRAC[1]	Fraction of the first speciation profile code to be used in combination (Real) (required)
G	PROF[2]	Second speciation profile code (Char 5) (required). See note on PROF[1].
H	FRAC[2]	Fraction of the second speciation profile code to be used in combination (Real)
I,K,etc	PROF[NPROF]	Third through NPROF speciation profile codes (Char 5) (requied). See note on PROF[1]
J,L,etc	FRAC[NPROF]	Fraction of the third through NPROF speciation profile code to be used in combination (Real)

An example of a two-entry GSPRO_COMBO file is as follows:

```
EXH__VOC 001001 1 3 8170 0.50 8171 0.20 8172 0.30
```

```
EVP__VOC 001001 1 3 8174 0.50 8175 0.20 8176 0.30
```

This file would do the following:

For county 01001 in period=1 (e.g., SPCMAT_PERIOD set to 1), Exhaust mode VOC would use 3 profiles (50% profile 8170, 20% profile 8171, and 30% profile 8172).

For county 01001 in period=1 (e.g., SPCMAT_PERIOD set to 1), Evaporative mode VOC would use 3 profiles (50% profile 8174, 20% profile 8175, and 30% profile 8176).

8.5.4. `GSREF`: Speciation cross-reference file

The speciation cross-reference file is used to match the emission sources with the speciation profiles in the GSPRO file. The GSREF file uses a header to define the point-source unique characteristics. The same file can be used for all source categories; the SMOKE programs use only the applicable entries in the file. The way that the file is applied in SMOKE ensures that the most-specific matches override the less-specific ones. A description of how the speciation cross-reference is applied to the inventory is provided in Section 6.15, "**Spcmat**" [286].

When the special keyword "COMBO" is placed in the speciation profile code column, all sources matching the cross-reference entry should use a combination speciation profile provided in Section 8.5.3, "GSPRO_COMBO: Combination speciation profiles file " [391]. Using the profiles listed in the GSPRO_COMBO

file, **Spcmat** can use the fractions and the information from the GSPRO and GSCNV for the profiles provied to create weighted profiles for the matching source.

8.5.4.1. Area- and mobile-source speciation cross-reference file entries

When emission type are used (as with mobile source processing using the **Movesmrg** program), column C must contain the emission types instead of the pollutants. For example, if the MEPROC file contained processes EXR (exhaust running), EPM (evaporative permeation), and CXR (crankcase running exhaust), then in order to use pollutant-specific entries in the GSREF file for mobile sources, records would need to be added to the file that use the emission types EXR__TOG, EPM__TOG, and CXR__TOG in column C. Note that the separator within these process-pollutant combinations must be a double underscore. The format of the GSREF file for area and mobile sources is shown in Table 8.41, "Format for area- and mobile-source GSREF" [393].

Table 8.41. Format for area- and mobile-source GSREF

Line	Position	Description
1+	A	Area: 20-digit SCC, or 10-digit SCC, or 7-digit SCC with zeros filling the remaining digits (Character) Mobile: 20-digit SCC code or 10-digit SCC (Character) Set to zero if cross-reference entry is not SCC-specific
	B	Speciation profile code (Character) (required; must match codes in the GSPRO file)
	C	Pollutant name or emission type (Character) (required; must match pollutants for profile code in GSPRO file; put zero id not pollutant-dependent)
	D	Country/state/county code (6-digit Character in format YSSCCC) (optional): • leave blank if not country/state-dependent • set to YSS000 for country/state-dependent • set to YSSCCC for county-dependent
	E	Maximum Achievable Technology Code (6-character String) (optional; use 0 if not needed)
	F	Standard Industrial Code (20-character String) (optional; use 0 if not needed)

8.5.4.2. Point-source speciation cross-reference file entries

The format of the GSREF file for point sources is shown in Table 8.42, "Format for point-source GS-REF" [394].

Table 8.42. Format for point-source `GSREF`

Line	Position	Description
1	A	/POINT DEFN/
	B	Number of point source characteristics in addition to plant ID in point source definition (for ORL, this will always be 4)
	C	Number of point source characteristic in addition to plant ID (if SCC is a part of the source definition); zero otherwise. • FF10 and ORL inventories: 4
2+	A	8- or 10- or 20- digit SCC (Character) (optional; set to zero if cross-reference entry is not SCC-specific)
	B	Speciation profile code (Character) (required; must match codes in the `GSPRO` file)
	C	Pollutant name or emission type (Character) (required; must match pollutants for profile code in `GSPRO` file; put zero if not pollutant-dependent)
	D	Country/state/county code (6-digit Character in format YSSCCC) (optional): • leave blank if not country/state-dependent • set to YSS000 for country/state-dependent • set to YSSCCC for county-dependent
	E	Maximum Achievable Technology Code (6-character String) (optional; use 0 if not needed)
	F	Standard Industrial Code (20-character String) (optional; use 0 if not needed)
	G	Facility ID for FF10 Point Sources (Plant ID for ORL), (Character(15), (optional ; leave blank if entry is not plant-specific)
	H	Characteristic 1 (for FF10, this is Unit ID; for ORL, this is Point ID), optional (Character(15))
	I	Characteristic 2 (for FF10, this is Release Point ID, (for ORL, this is stack ID); optional (Character(15))
	J	Characteristic 3 (for FF10, this is Process ID, (for ORL, this is segment ID); optional (Character(15))
	K	Characteristic 4 (for FF10 or ORL, this is 10-digit SCC, or 5-digit with zeroes, or 20-digit maximum SCC) (Character) (optional)
	L	Characteristic 5 (blank for FF10 or ORL) (Character) (optional)

8.5.5. `GSTAG`: Speciation tagging cross-reference file

The speciation tagging cross-reference file allows **Spcmat** to tag existing model species based on the match between the source characteristics listed in the GSTAG file and source characteristics for each source in the inventory. The **Spcmat** program will append the characters of the tag label specified in the GSTAG file. Note that this approach will not address raw inventory pollutants (VOC, NOX, PM2_5 and PM10) or any other model species from biogenic sources from the BEIS model. When the environment variable SRC_TAGGING is set to "Y", **Spcmat** will look for the GSTAG file. The default setting for the

SRC_TAGGING variable is "N" to support back-ward comparability. If an invalid character appears in a tag label in the GSTAG file, such as "#" or "+" is used, SMOKE will return an error.

The format of the GSTAG file is shown in Table 8.43, "Format for GSTAG" [395].

Table 8.43. Format for GSTAG

Line	Position	Description
1+	A	Country/state/county code (6-digit Character in format YSSCCC) (optional): • leave blank if not country/state-dependent • set to YSS000 for country/state-dependent • set to YSSCCC for county-dependent
	B	20-digit maximum SCC (Character) (optional; set to zero if cross-reference entry is not SCC-specific)
	C	Species name (Character) (required; must match Species in GSPRO file; 16-character maximum). If missing or -9, SMOKE willl return an error.
	D	Tagging label (Character) (required; 8-character maximum. The combination of Species name in column C and the tagging label can not exceed 16 chracters)
	E	Maximum Achievable Technology Code (6-character String) (optional; use 0 if not needed)
	F	Standard Industrial Code (20-character String) (optional; use 0 if not needed)
	G	Plant ID for point sources (Character) (optional; leave bank or -9 if entry is not plant-specific)

An example of a two-entry GSTAG file is as follows:

001001, 20400408, ALD2, _T1

001001, 2530050000, NO, _T1

This file would do the following:

ALD2 species from County 0100 and SCC 20400408 is tagged as ALD2_T1.

NO species from County 0100 and SCC 2530050000 is tagged as NO_T1.

8.6. Growth and Controls Files

The directories where the growth and controls files should be stored are listed in Table 8.5, "Growth and Controls Input File Locations" [348]. The table also lists the programs that use each input file.

8.6.1. GCNTL: Files for controls and/or past- and future-year growth

SMOKE uses growth and control packets to get the information needed for growth and control of emissions. A packet is a group of inputs used to apply specific types of controls. A single control file containing a

series of packets can be shared by area, mobile, and point sources and the packets that do not apply to a given source category will be ignored. The following packets are recognized by the SMOKE control programs:

- /ALLOWABLE/ contains county-specific, SIC-specific, SCC-specific controls, caps, and replacement emissions. Contributes to the multiplicative control matrix. Section 8.6.1.1, "/ALLOWABLE/ Packet" [396]

- /CONTROL/ contains settings for control efficiency, rule effectiveness, and rule penetration. Contributes to the multiplicative control matrix. Section 8.6.1.2, "/CONTROL/ Packet" [397]

- /CTG/ contains settings for CTG, MACT, and RACT controls. Contributes to the multiplicative control matrix. Section 8.6.1.3, "/CTG/ Packet" [398]

- /MACT/ contains settings to support the primary MACT controls needed for toxics processing. Contributes to the multiplicative control matrix. Section 8.6.1.4, "/MACT/ Packet" [399]

- /PROJECTION/ contains settings to permit growth of emission values into the past or future. Creates a growth matrix for each pollutant and/or activity. Section 8.6.1.5, "/PROJECTION/ Packet" [400]

- /REACTIVITY/ contains settings needed for reactivity-based controls. Creates the reactivity control matrix. Section 8.6.1.6, "/REACTIVITY/ Packet" [401]

The formats for these packets are described in the section below. All of the packets are list-directed.

8.6.1.1. /ALLOWABLE/ Packet

The format of the /ALLOWABLE/ packet is shown in Table 8.44, "Format for /ALLOWABLE/ packet in `GCNTL`" [397].

Table 8.44. Format for /ALLOWABLE/ packet in `GCNTL`

Line	Position	Description
1	A	/ALLOWABLE/
2+	A	Country/State/County code or Country/state code with zero for county, or zero (Character)
	B	8 or 10-digit SCC, (20 digit maximum) or pieces of SCC with remaining digits filled in with zeros, or zero (Character)
	C	Pollutant ID, or -9 if not a pollutant-specific control; in most cases, the cap or replacement value will be a pollutant-specific value, and that pollutant name needs to be placed in this column (Character)
	D	Control factor (Real) (no longer used by SMOKE, enter -9 as placeholder)
	E	Allowable emissions cap value [short tons/day] (Real) (required if no "replace" emissions are given)
	F	Allowable emissions replacement value [short tons/day] (Real) (required if no "cap" emissions are given)
	G	SIC for point sources or -9 if not needed (Character(20)) (not supported by SMOKE at this time)
	H	Facility ID for FF10 Point Sources (Plant ID for ORL), optional (Character(15))
	I	Characteristic 1 (for FF10, this is Unit ID; for ORL, this is Point ID), optional (Character(15))
	J	Characteristic 2 (for FF10, this is Release Point ID, (for ORL, this is stack ID); optional (Character(15))
	K	Characteristic 3 (for FF10, this is Process ID, (for ORL, this is segment ID); optional (Character(15))
	L	Characteristic 4 (blank for FF10 or ORL inventory input format), optional (Character)
	M	Characteristic 5 (blank for FF10 or ORL inventory input format), optional (Character)
3	A	/END/

8.6.1.2. /CONTROL/ Packet

The format of the /CONTROL/ packet is shown in Table 8.45, "Format for /CONTROL/ packet in `GCNTL`" [398].

Table 8.45. Format for /CONTROL/ packet in `GCNTL`

Line	Position	Description
1	A	/CONTROL/
2+	A	Country/State/County code or Country/state code with zero for county, or zero (Character)
	B	8 or 10-digit SCC (20 digit maximum), or pieces of SCC with remaining digits filled in with zeros, or zero (Character)
	C	Pollutant ID, or -9 if not a pollutant-specific control (Character)
	D	Primary control equipment code (PCEC); -9 or zero applies to all equipment (not yet supported)
	E	Control efficiency (value should be a percent; e.g., enter 90 for a 90% control efficiency) (Real)
	F	Rule effectiveness (value should be a percent; e.g., enter 50 for a 50% rule effectiveness) (Real)
	G	Rule penetration rate (value should be a percent; e.g., enter 80 for a 80% rule penetration) (Real)
	H	Standard Industrial Category (SIC), optional (20-digit Character)
	I	Maximum Achievable Control Technology (MACT) code, optional (6-character string)
	J	Application control flag (Character) • Y = control should be applied to inventory • N = control will not be used
	K	Replacement flag (Character) • A = control is applied in addition to controls from /MACT/ packet • R = control replaces controls from /MACT/ packet
	L	Facility ID for FF10 Point Sources (Plant ID for ORL), optional (Character(15))
	M	Characteristic 1 (for FF10, this is Unit ID; for ORL, this is Point ID), optional (Character(15))
	N	Characteristic 2 (for FF10, this is Release Point ID, (for ORL, this is stack ID); optional (Character(15))
	O	Characteristic 3 (for FF10, this is Process ID, (for ORL, this is segment ID); optional (Character(15))
	P	Characteristic 4 (blank for FF10 or ORL inventory input format), optional (Character)
	Q	Characteristic 5 (blank for FF10 or ORL inventory input format), optional (Character)
3	A	/END/

8.6.1.3. /CTG/ Packet

The format of the /CTG/ packet is shown in Table 8.46, "Format for /CTG/ packet in `GCNTL`" [399].

Table 8.46. Format for /CTG/ packet in GCNTL

Line	Position	Description
1	A	/CTG/
2+	A	Country/State/County code or Country/state code with zero for county, or zero (Character)
	B	8 or 10-digit SCC (20 digit maximum), or pieces of SCC with remaining digits filled in with zeros, or zero (Character)
	C	Pollutant ID, or -9 if not a pollutant-specific control; in most cases, the emissions cutoff value will be a pollutant-specific value, and that pollutant name needs to be placed in this column (Character)
	D	Control technology control factor; value should be a fraction 0 to 1. (Real)
	E	CTG emissions cutoff for application of CTG, MACT, or RACT factor (short tons/day) (Real)
	F	MACT control factor; value should be a fraction 0 to 1, or -9 if missing (Real)
	G	RACT; value should be a fraction 0 to 1; or -9 if missing (Real)
3	A	/END/

8.6.1.4. /MACT/ Packet

The format of the /MACT/ packet is shown in Table 8.47, "Format for /MACT/ packet in GCNTL" [400].

Table 8.47. Format for /MACT/ packet in `GCNTL`

Line	Position	Description
1	A	/MACT/
2+	A	Maximum Achievable Control Technology (MACT) code, optional (6-character string)
	B	up to 20-digit maximum, or 8 or 10-digit SCC, or pieces of SCC with remaining digits filled in with zeroes or 0 (Character)
	C	Source type (2-character string) • 00 = both • 01 = major • 02 = CAA Section 12 area sources • 03 = nonroad • 04 = onroad The source type is only used to make a final determination about whether the entry applies for a source. If the record matches the source otherwise (MACT or SCC matches) but the source type does not match, then the source receives no controls from this entry.
	D	Application control flag (Character) • Y = control should be applied to inventory • N = control will not be used
	E	Pollutant ID, or -9 if not a pollutant-specific control (Character)
	F	Control efficiency of existing sources (value should be 0-100) (Real)
	G	Control efficiency of new sources (value should be 0-100) (Real)
	H	Fraction of new and existing sources (value should be 0-100) (Real)
3	A	/END/

8.6.1.5. /PROJECTION/ Packet

The format of the /PROJECTION/ packet is shown in Table 8.48, "Format for /PROJECTION/ packet in `GCTNL`" [401]. Note that if both the SCC and SIC is supplied in a given packet entry, the SCC will be used for source matching.

Table 8.48. Format for /PROJECTION/ packet in GCTNL

Line	Position	Description
1	A	/PROJECTION <4-digit from year> <4-digit to year>/
2+	A	Country/State/County code, or Country/state code with zero for county, or zero (Character)
	B	up to 20 digit maximum SCC, or 8 or 10-digit SCC, or pieces of SCC with remaining digits filled in with zeros, or zero (Character)
	C	Projection factor [enter number on fractional basis; e.g., enter 1.2 to increase emissions by 20%] (Real)
	D	Pollutant ID, or -9 if not a pollutant-specific entry
	E	SIC, or 2-digit with remaining digits filled in with zeroes or blank for missing, optional (Character(20))
	F	Maximum Achievable Control Technology code (MACT), optional (6-character string)
	G	Facility ID for FF10 Point Sources (Plant ID for ORL), optional (Character(15))
	H	Characteristic 1 (for FF10, this is Unit ID; for ORL, this is Point ID), optional (Character(15))
	I	Characteristic 2 (for FF10, this is Release Point ID, (for ORL, this is stack ID); optional (Character(15))
	J	Characteristic 3 (for FF10, this is Process ID, (for ORL, this is segment ID); optional (Character(15))
	K	Characteristic 4 (blank for FF10 or ORL inventory input format), optional (Character)
	L	Characteristic 5 (blank for FF10 or ORL inventory input format), optional (Character)
3	A	/END/

8.6.1.6. /REACTIVITY/ Packet

The format of the /REACTIVITY/ packet is shown in Table 8.49, "Format for /REACTIVITY/ packet in GCNTL" [402].

Table 8.49. Format for /REACTIVITY/ packet in `GCNTL`

Line	Position	Description
1	A	/REACTIVITY <4-digit base year> <4-digit future year>/
2+	A	Country/State/County code or Country/state code with zero for county, or zero (Character)
	B	20 digit maximum SCC, or 8 or 10-digit SCC, or pieces of SCC with remaining digits filled in with zeros, or zero (Character)
	C	Pollutant ID, required (Character)
	D	Replacement emissions for base year [short tons/day] (Real)
	E	Projection factor (or 0. for none; enter fractional value; e.g., 1.2 to increase emissions by 20%) (Real)
	F	Future year 10-digit (up to a maximum of 20 digit) SCC code (or zero to set equal to base case) (Character)
	G	Speciation profile number (or zero to set equal to base case) (Real)
	H	Market penetration of new speciation [fraction/year; e.g., enter 0.2 for 20% market penetration per year] (0. to 1.0) (Real)
	I	SIC for point sources or -9 if not needed; not supported by SMOKE at this time (Character(20)
	J	Facility ID for FF10 Point Sources (Plant ID for ORL), optional (Character(15))
	K	Characteristic 1 (for FF10, this is Unit ID; for ORL, this is Point ID), optional (Character(15))
	L	Characteristic 2 (for FF10, this is Release Point ID, (for ORL, this is stack ID); optional (Character(15))
	M	Characteristic 3 (for FF10, this is Process ID, (for ORL, this is segment ID); optional (Character(15))
	N	Characteristic 4 (blank for FF10 and ORL inventory input format), optional (Character)
	O	Characteristic 5 (blank for FF10 and ORL inventory input format), optional (Character)
3	A	/END/

8.7. Special Merging Files

The directories where the special merging files should be stored are listed in Table 8.6, "Special Merging Input File Locations" [348]. The table also lists the programs that use each input file.

8.7.1. `FILELIST`: Gridded merge 2-D file list

The `FILELIST` file is used to provide a list of logical file names to the **Mrggrid** program. This list should contain the logical file names for all 2- or 3-D files to be merged in a given run. It is a very simple file, and has no limit to the number of files listed. Note that unlike most list files used in SMOKE, `FILELIST` requires a list of logical file names, *not* the actual file name. The format of the `FILELIST` file is shown in Table 8.50, "Format for `FILELIST`" [403].

Table 8.50. Format for `FILELIST`

Line	Position	Description
1+	A	Logical file name

8.8. Meteorology Files

The directories where the meteorology files should be stored are listed in Table 8.7, "Meteorology Input File Locations" [349]. The table also lists the programs that use each input file.

The five meteorology files that SMOKE needs for biogenic emissions adjustments, calculation of on-road mobile source emission, and plume rise calculations are `GRID_CRO_2D`, `GRID_CRO_3D`, `MET_CRO_2D`, `MET_CRO_3D`, and `MET_DOT_3D`. The files are usually based on MM5 data but must be specially formatted by a postprocessor like MCIP or MCPL.

`GRID_CRO_2D` - 2-D grid parameters file

`GRID_CRO_3D` - 3-D grid parameters file

`MET_CRO_2D` - 2-D cross-point data file (surface meteorology data)

`MET_CRO_3D` - 3-D cross-point data file (layered meteorology data)

`MET_DOT_3D` - 3-D dot-point data file (layered meteorology data)

Table 8.51. Meteorology variables used by program

File name	Program	Variable	Description	Notes
GRID_CRO_2D	**Tmpbeis3**	LAT	grid cell latitude (degrees)	
		LON	grid cell longitude (degrees)	
		LON	grid cell longitude (degrees)	only used if BG_CLOUD_TYPE = 1
GRID_CRO_3D	**Laypoint**	X3HT0F	height of layer face above ground (m)	only used when layer structure is static
		X3HT0M	height of layer middle above ground (m)	only used when layer structure is static
MET_CRO_2D	**Laypoint**	HFX	sensible heat flux (watts/m^2)	
		PBL	PBL height (m)	
		PRSFC	surface pressure (Pascals)	
		TEMP1P5	temperature at 1.5 m (Kelvins)	
		USTAR	average friction velocity (m/s)	
	Met4moves	TEMP2	temperature at 2m (Kelvins)	variable name is set by TVARNAME, all **Met4moves** variables must be in same file
	Tmpbeis3	RGRND	solar radiation reaching surface (watts/m^2)	variable name is set by RAD_VAR, actual file used is determined by BIO-MET_SAME
		RGRND	solar radiation reaching surface (watts/m^2)	only used if BG_CLOUD_TYPE = 1, variable name is set by RAD_VAR, actual file used is determined by BIO-MET_SAME
MET_CRO_3D	**Laypoint**	DENS	total air density (kg/m^3)	
		PRES	pressure (Pascals)	
		QV	water vapor mixing ratio (kg/kg)	

File name	Program	Variable	Description	Notes
		TA	air temperature (Kelvins)	
		ZF	full-layer height above ground (m)	
		ZH	mid-layer height above ground (m)	
	Met4moves	PRES	pressure (Pascals)	all **Met4moves** variables must be in same file
	Met4moves	QV	water vapor mixing ratio (kg/kg)	all **Met4moves** variables must be in same file
	Tmpbeis3	PRES	pressure (Pascals)	variable name is set by `PRES_VAR`
		RC	convective rainfall (cm)	only used if `BEIS_VERSION` = 3.14, variable name is set by `RC_VAR`
		RN	nonconvective rainfall (cm)	only used if `BEIS_VERSION` = 3.14, variable name is set by `RN_VAR`
		SLTYP	soil texture type by USDA category	only used if `BEIS_VERSION` = 3.14 and `PX_VERSION` = Y, variable name is set by `IS-LTYP_VAR`
		SOIM1	volumetric soil moisture in top centimeter (m^3/m^3)	only used if `BEIS_VERSION` = 3.14 and `PX_VERSION` = Y, variable name is set by `SOILM_VAR`
		SOIT1	soil temperature in top centimeter (Kelvins)	only used if `BEIS_VERSION` = 3.14 and `PX_VERSION` = Y, variable name is set by `SOILT_VAR`
		TA	air temperature (Kelvins)	variable name is set by `TMPR_VAR`, actual file opened depends on `MET_FILE1`

File name	Program	Variable	Description	Notes
	Tmpbeis3	TA	air temperature (Kelvins)	variable name is set by `TMPR_VAR`, actual file opened depends on `MET_FILE1`/`METLIST`
`MET_DOT_3D`	**Laypoint**	UWIND	U-component of wind (m/s)	
		VWIND	V-component of wind (m/s)	

8.8.1. `METLIST`: List of meteorology files

The `METLIST` file is used to provide the file names of available meteorology files to the **Met4moves** programs. **Met4moves** runs once for the entire modeling episode, and therefore needs all of the files at one time. The meteorology files provided to **Met4moves** must cover the desired episode and contain the requested variables. In the example SMOKE scripts, this file is generated automatically. The format of this file is shown in Table 8.52, "Format for `METLIST`" [406].

Table 8.52. Format for `METLIST`

Line	Position	Description
1+	A	Full name and path of meteorology file

8.9. Source-Specific files

8.9.1. Biogenics

The directories where the biogenic-specific input files should be stored are listed in Table 8.8, "Biogenic-Specific Input File Locations" [349]. The table also lists the programs that use each input file.

8.9.1.1. `BIOSEASON`: Bioseason switch file

The `BIOSEASON` switch file is an I/O API netCDF file used to indicate how each day in a given year should be treated, either as summer or winter. This file can be created using the **Metscan** utility program. The `BIOSEASON` file is time-dependent and usually contains data for an entire year (365 or 366 days). It uses one variable, SEASON, which is either 0 (grid cell should use winter factors for current day) or 1 (grid cell should use summer factors for current day).

8.9.1.2. `B360FAC`: Biogenic emissions factors file for BEIS v3.60

The `B360FAC` file for BEIS version 3.60 contains LAI, dry leaf biomass, winter biomass factor, indicator of specific leaf weight, and normalized emission fluxes for 34 different species/compounds listed in Table 8.53, "Format for `B3FAC`, BEIS version 3.60" [407].

Table 8.53. Format for `B3FAC`, BEIS version 3.60

Line	Position	Description
1+	A	Abbreviated name for land use type (Character)
	B	String indicating type (e.g. Leaf Area Index (LAI), Dry Leaf Biomass (LFBIO), Winter Biomass (WFAC), Specific Leaf Weight (SLW), Isoprene (ISOP), Sesquiterpene (SESQT) emissions, etc.) (Character)
	C	Units or "none" if no units (Character)
	D	Value of LAI, LFBIO, WFAC, SLW, or Normalized emissions (i.e., ISOP, SESQT, and son on) flux for item in position B (g C/km^2-hr) (Real)

8.9.1.3. `B3FAC`: Biogenic emissions factors file for BEIS v3.14 or earlier

The `B3FAC` file for BEIS version 3.14 contains LAI, dry leaf biomass, winter biomass factor, indicator of specific leaf weight, and normalized emission fluxes for 34 different species/compounds listed in Table 8.54, "Format for `B3FAC`, BEIS version 3.14" [408].

Table 8.54. Format for `B3FAC`, BEIS version 3.14

Line	Position	Description
1+	1-16	Abbreviated name for land use type (Character)
	A	Leaf area index (m^2/m^2) (Integer)
	B	Dry leaf biomass (g/m^2) (Integer)
	C	Winter biomass factor (Real)
	D	Indicator of specific leaf weight (Integer)
	E	Normalized emissions flux for ISOP (g C/km^2-hr) (Real)
	F	Normalized emissions flux for MBO (g C/km^2-hr) (Real)
	G	Normalized emissions flux for METH (g C/km^2-hr) (Real)
	H	Normalized emissions flux for APIN (g C/km^2-hr) (Real)
	I	Normalized emissions flux for BPIN (g C/km^2-hr) (Real)
	J	Normalized emissions flux for D3CAR (g C/km^2-hr) (Real)
	K	Normalized emissions flux for DLIM (g C/km^2-hr) (Real)
	L	Normalized emissions flux for CAMPH (g C/km^2-hr) (Real)
	M	Normalized emissions flux for MYRC (g C/km^2-hr) (Real)
	N	Normalized emissions flux for ATERP (g C/km^2-hr) (Real)
	O	Normalized emissions flux for BPHE (g C/km^2-hr) (Real)
	P	Normalized emissions flux for SABI (g C/km^2-hr) (Real)
	Q	Normalized emissions flux for PCYM (g C/km^2-hr) (Real)
	R	Normalized emissions flux for OCIM (g C/km^2-hr) (Real)
	S	Normalized emissions flux for ATHU (g C/km^2-hr) (Real)
	T	Normalized emissions flux for TRPO (g C/km^2-hr) (Real)
	U	Normalized emissions flux for GTERP (g C/km^2-hr) (Real)
	V	Normalized emissions flux for ETHE (g C/km^2-hr) (Real)
	W	Normalized emissions flux for PROPE (g C/km^2-hr) (Real)
	X	Normalized emissions flux for ETHO (g C/km^2-hr) (Real)
	Y	Normalized emissions flux for ACET (g C/km^2-hr) (Real)
	Z	Normalized emissions flux for HEXA (g C/km^2-hr) (Real)
	AA	Normalized emissions flux for HEXE (gC/km^2-hr) (Real)
	AB	Normalized emissions flux for HEXY (g C/km^2-hr) (Real)
	AC	Normalized emissions flux for FORM (g C/km^2-hr) (Real)
	AD	Normalized emissions flux for ACTAL (gC/km^2-hr) (Real)
	AE	Normalized emissions flux for BUTE (g C/km^2-hr) (Real)
	AF	Normalized emissions flux for ETHA (g C/km^2-hr) (Real)
	AG	Normalized emissions flux for FORAC (g C/km^2-hr) (Real)
	AH	Normalized emissions flux for ACTAC (g C/km^2-hr) (Real)
	AI	Normalized emissions flux for BUTO (g C/km^2-hr) (Real)
	AJ	Normalized emissions flux for CO (g C/km^2-hr) (Real)

Line	Position	Description
	AK	Normalized emissions flux for ORVOC (g C/km^2-hr) (Real)
	AL	Normalized emissions flux for NO (g N/km^2-hr) (Real)

8.9.1.4. `BELD4`: Landuse input file for BEIS v3.60

Table 8.55, "Land use categories in BELD4" [410] contains names and descriptions of the 286 land use categories in BELD4.

Table 8.55. Land use categories in BELD4

Land use category	Description
NLCD_11	Open Water
NLCD_12	Perennial Ice/Snow
NLCD_21	Developed Open Space
NLCD_22	Developed Low Intensity
NLCD_23	Developed Medium Intensity
NLCD_24	Developed High Intensity
NLCD_31	Barren Land (Rock/Sand/Clay)
NLCD_32	Unconsolidated Shore
NLCD_41	Deciduous Forest
NLCD_42	Evergreen Forest
NLCD_43	Mixed Forest
NLCD_51	Dwarf Scrub
NLCD_52	Shrub/Scrub
NLCD_71	Grassland/Herbaceous
NLCD_72	Sedge/Herbaceous
NLCD_73	Lichens
NLCD_74	Moss
NLCD_75	Tundra
NLCD_81	Pasture/Hay
NLCD_82	Cultivated Crops
NLCD_90	Woody Wetlands
NLCD_91	Palustrine Forested Wetland
NLCD_92	Palustrine Scrub/Shrub Wetland
NLCD_93	Estuarine Forested Wetland
NLCD_94	Estuarine Scrub/Shrub Wetland
NLCD_95	Emergent Herbaceous Wetlands
NLCD_96	Palustrine Emergent Wetland
NLCD_97	Estuarine Emergent Wetland
NLCD_98	Palustrine Aquatic Bed
NLCD_99	Estuarine Aquatic Bed
MODIS_0	water
MODIS_1	Evergreen needleleaf forest
MODIS_2	Evergreen broadleaf forest
MODIS_3	Deciduous needleleaf forest
MODIS_4	Deciduous broadleaf forest
MODIS_5	Mixed forests
MODIS_6	Closed shrublands

Land use category	Description
MODIS_7	Open shrublands
MODIS_8	Woody savannas
MODIS_9	Savannas
MODIS_10	Grasslands
MODIS_11	Permanent wetlands
MODIS_12	Croplands
MODIS_13	Urban and built up
MODIS_14	Cropland / natural vegetation mosaic
MODIS_15	Permanent snow and ice
MODIS_16	Barren or sparsely vegetated
MODIS_17	IGBP water
MODIS_254	Unclassified
MODIS_255	Fill value (normally ocean water)
Alfalfa	Alfalfa
Acacia	Acacia spp
Ailanthus	Ailanthus altissima
Alder	Alnus
Alfalfa	Alfalfa irrigated
Alfalfa_ir	Malus
Apple	Malus
Ash	Fraxinus
Barley	Barley
Barley_ir	Barley irrigated
Basswood	Tilia
Beans	Beans
Beans_ir	Beans irrigated
BeansEdible	Beans edible
BeansEdible_ir	Beans edible irrigated
Beech	Fagus
Birch	Betula
Bumelia_gum	Bumelia lanuginosa
Cajeput	Melaleuca quinquenervia
Califor_laurel	Umbellularia californica
Cascara_buckthor	Rhamnus purshiana
Canola	Canola
Canola_ir	Canola irrigated
Cascara_buckthor	Rhamnus purshiana
Castanea	American chestnut or chinkapin

Land use category	Description
Catalpa	Catalpa
Cedar_chamaecyp	Orford, Alaska, and Atlantic white cedars
Cedar_thuja	Northern white and western red cedars
Chestnut_buckeye	Aesculus - Horse chestnut or buckeyes
Chinaberry	Melia azedarach
CornGrain	Corn Grain
CornGrain_ir	Corn Grain irrigated
CornSilage	Corn Silage
CornSilage_ir	Corn Silage irrigated
Cotton	Cotton
Cotton_ir	Cotton irrigated
Cypress_cupress	Cupressus spp
Cypress_taxodium	Taxodium - bald and pond cypress and yew
Dogwood	Cornus
Douglas_fir	Pseudotsuga
East_hophornbean	Ostrya virginiana
Elder	Sambucus
Elm	Ulmus
Eucalyptus	Eucalyptus spp
Fir_balsam	Abies balsamea
Fir_CA_red	Abies magnifica
Fir_corkbark	Abies lasiocarpa
Fir_fraser	Abies fraseri
Fir_grand	Abies grandis
Fir_noble	Abies procera
Fir_Pacf_silver	Abies amabilis
Fir_SantaLucia	Abies bracteata
Fir_Shasta_red	Abies magnifica shastensis
Fir_spp	Abies spp
Fir_subalpine	Abies lasiocarpa arizonica
Fir_white	Abies concolor
Gleditsia_locust	Waterlocust and Honeylocust
Hackberry	Celtis
Hawthorn	Crataegus
Hay	Hay
Hay_ir	Hay irrigated
Hemlock	Tsuga
Hickory	Carya - includes pecan

Land use category	Description
Holly_American	Ilex
Hornbeam	Carpinus caroliniana
Incense_cedar	Calocedrus decurrens
Juniper	Juniperus - include Sou/Eas red cedar
KY_coffeetree	Gymnocladus dioicus
Larch	Larix
Loblolly_bay	Gordonia lasianthus
Madrone	Arbutus
Magnolia	Magnolia
Mahogany	Cercocarpus
Maple_bigleaf	Acer grandidentatum
Maple_bigtooth	Acer macrophyllum
Maple_black	Acer nigrum
Maple_boxelder	Acer negundo
Maple_FL	Acer barbatum
Maple_mtn	Acer spicatum
Maple_Norway	Acer platanoides
Maple_red	Acer rubrum
Maple_RkyMtn	Acer glabrum
Maple_silver	Acer saccharinum
Maple_spp	Acer spp
Maple_striped	Acer pensylvanicum
Maple_sugar	Acer saccharum
Mesquite	Prosopis
Misc_hardwoods	Misc hardwoods - undefined from FIA
Mixed_conifer_sp	Mixed conifers - underfined from FIA
Mountain_ash	Sorbus
Mulberry	Morus
Nyssa	Tupelo and blackgum
Oak_AZ_white	Quercus arizonica
Oak_bear	Quercus ilicifolia
Oak_black	Quercus velutina
Oak_blackjack	Quercus marilandica
Oak_blue	Quercus douglasii
Oak_bluejack	Quercus incana
Oak_bur	Quercus macrocarpa
Oak_CA_black	Quercus kelloggii
Oak_CA_live	Quercus agrifolia

Land use category	Description
Oak_CA_white	Quercus lobata
Oak_canyon_live	Quercus chrysolepis
Oak_chestnut	Quercus prinus
Oak_chinkapin	Quercus muehlenbergii
Oak_delta_post	Quercus stellata
Oak_Durand	Quercus durandii
Oak_Emery	Quercus emoryi
Oak_Engelmann	Quercus engelmannii
Oak_evergreen_sp	Quercus spp
Oak_Gambel	Quercus gambelii
Oak_interio_live	Quercus wislizenni
Oak_laurel	Quercus laurifolia
Oak_live	Quercus virginiana
Oak_Mexicanblue	Quercus oblongifolia
Oak_Northrn_pin	Quercus ellipsoidalis
Oak_Northrn_red	Quercus rubra
Oak_nuttall	Quercus nuttallii
Oak_OR_white	Quercus garryana
Oak_overcup	Quercus lyrata
Oak_pin	Quercus palustris
Oak_post	Quercus stellata
Oak_scarlet	Quercus coccinea
Oak_scrub	Quercus spp
Oak_shingle	Quercus imbricaria
Oak_Shumrd_red	Quercus shumardii
Oak_silverleaf	Quercus hypoleucoides
Oak_Southrn_red	Quercus falcata
Oak_spp	Quercus spp
Oak_swamp_cnut	Quercus michauxii
Oak_swamp_red	Quercus falcata
Oak_swamp_white	Quercus bicolor
Oak_turkey	Quercus laevis
Oak_water	Quercus nigra
Oak_white	Quercus alba
Oak_willow	Quercus phellos
Oaks	Oaks
Oaks_ir	Oaks irrigated
Osage_orange	Maclura pomifera

Land use category	Description
Other_crop	Other crops
Other_crop_ir	Other crops irrigated
Other_grass	Other grass
Other_grass_ir	Other grass irrigated
Paulownia	Paulownia tomentosa
Pawpaw	Asimina triloba
Persimmon	Diospyros virginiana
Pine_Apache	Pinus engelmannii
Pine_Austrian	Pinus nigra
Pine_AZ	Pinus ponderosa
Pine_Bishop	Pinus muricata
Pine_blackjack	Pinus ponderosa
Pine_brstlcone	Pinus aristata
Pine_chihuahua	Pinus leiophylla
Pine_Coulter	Pinus coulteri
Pine_digger	Pinus sabiniana
Pine_Ewhite	Pinus strobus
Pine_foxtail	Pinus balfouriana
Pine_jack	Pinus banksiana
Pine_Jeffrey	Pinus jefferyi
Pine_knobcone	Pinus attenuata
Pine_limber	Pinus flexilis
Pine_loblolly	Pinus taeda
Pine_lodgepole	Pinus contorta
Pine_longleaf	Pinus palustris
Pine_Monterey	Pinus radiata
Pine_pinyon	Pinus monophylla
Pine_pinyon_brdr	Pinus discolor
Pine_pinyon_cmn	Pinus edulis
Pine_pitch	Pinus rigida
Pine_pond	Pinus serotina
Pine_ponderosa	Pinus ponderosa
Pine_red	Pinus resinosa
Pine_sand	Pinus clausa
Pine_scotch	Pinus sylvestris
Pine_shortleaf	Pinus echinata
Pine_slash	Pinus elliottii
Pine_spruce	Pinus glabra

Land use category	Description
Pine_sugar	Pinus lambertiana
Pine_Swwhite	Pinus strobiformis
Pine_tablemtn	Pinus pungens
Pine_VA	Pinus virginiana
Pine_Washoe	Pinus washoensis
Pine_whitebark	Pinus albicaulis
Pine_Wwhite	Pinus monticola
Pine_yellow	Pinus ponderosa
Populus	Cottonwood, poplars, and aspens
Potatoes	Potatoes
Potatoes_ir	Potatoes irrigated
Prunus	Cherry and plum
Redbud	Cercis
Rice	Rice
Rice_ir	Rice irrigated
Robinia_locust	Black and NM locust
Rye	Rye
Rye_ir	Rye irrigated
Sassafras	Sassafras albidum
Sequoia	Redwood and giant sequoia
Serviceberry	Amelanchier arborea
Silverbell	Halesia
Smoketree	Cotinus obovatus
Soapberry_westrn	Sapindus drummondii
SorghumGrain	Sorghum Grain
SorghumGrain_ir	Sorghum Grain irrigated
SorghumSilage	Sorghum Silage
SorghumSilage_ir	Sorghum Silage irrigated
Sourwood	Oxydendrum arboretum
Soybeans	Soybeans
Soybeans_ir	Soybeans irrigated
Sparkleberry	Vaccinium arboreum
Spruce_black	Picea marianna
Spruce_blue	Picea pungens
Spruce_Brewer	Picea breweriana
Spruce_Englemann	Picea engelmannii
Spruce_Norway	Picea abies
Spruce_red	Picea rubens

Land use category	Description
Spruce_Sitka	Picea sitchensis
Spruce_spp	Picea spp
Spruce_white	Picea glauca
Sweetgum	Liquidambar styraciflua
Sycamore	Platanus
Tallowtree_chins	Sapium sebiferum
Tamarix	Tamarix chinensis
Tanoak	Lithocarpus densiflorus
Torreya	Torreya californica
Tung_oil_tree	Aleurites fordii
Unknown_tree	Unknown spp
Walnut	Juglans - includes butternut
Wheat_spring	Wheat spring
Wheat_spring_ir	Wheat spring irrigated
Wheat_winter	Wheat winter
Wheat_winter_ir	Wheat winter irrigated
Water_elm	Planera aquatica
Willow	Salix
Yellow_poplar	Liriodendron tulipifera
Yellowwood	Cladrastis lutea
Yucca_Mojave	Yucca schidigera

8.9.1.5. `BELD3`: Landuse input file for BEIS v3.14 or earlier

Table 8.56, "Land use categories in BELD3" [418] contains names and descriptions of the 230 land use categories in BELD3.

Table 8.56. Land use categories in BELD3

Land use category	Description
USGS_urban	USGS urban and built up land
USGS_drycrop	USGS dryland cropland and pasture
USGS_irrcrop	USGS irrigated cropland and pasture
USGS_cropgrass	USGS cropland/grassland mosaic
USGS_cropwdlnd	USGS cropland/woodland mosaic
USGS_grassland	USGS grassland
USGS_shrubland	USGS shrubland
USGS_shrubgrass	USGS mixed shrubland/grassland
USGS_savanna	USGS savanna
USGS_decidforest	USGS deciduous broadleaf forest
USGS_evbrdleaf	USGS evergreen broadleaf forest
USGS_coniferfor	USGS evergreen needleleaf forest
USGS_mxforest	USGS mixed forest
USGS_water	USGS water bodies
USGS_wetwoods	USGS wooded wetland
USGS_sprsbarren	USGS barren or sparsely vegetated
USGS_woodtundr	USGS wooded tundra
USGS_mxtundra	USGS mixed tundra
USGS_snowice	USGS snow or ice
Alfalfa	Alfalfa
Barley	Barley
CornGrain	CornGrain
Corn_ir	CornGrain irrigated
Cotton	Cotton
Grass	Grass (from ag data)
Hay	Hay
Misc_crop	Miscellaneous crops
Oats	Oats
Pasture	Pasture (from ag data)
Peanuts	Peanuts
Potatoes	Potatoes
Rice	Rice
Rye	Rye
Sorghum	Sorghum
Soybeans	Soybeans
Tobacco	Tobacco
Wheat	Wheat

Land use category	Description
Acacia	Acacia spp
Ailanthus	Ailanthus altissima
Alder	Alnus
Apple	Malus
Ash	Fraxinus
Basswood	Tilia
Beech	Fagus
Birch	Betula
Bumelia_gum	Bumelia lanuginosa
Cajeput	Melaleuca quinquenervia
Califor_laurel	Umbellularia californica
Cascara_buckthor	Rhamnus purshiana
Castanea	American chestnut or chinkapin
Catalpa	Catalpa
Cedar_chamaecyp	Orford, Alaska, and Atlantic white cedars
Cedar_thuja	Northern white and western red cedars
Chestnut_buckeye	Aesculus - Horse chestnut or buckeyes
Chinaberry	Melia azedarach
Cypress_cupress	Cupressus spp
Cypress_taxodium	Taxodium - bald and pond cypress and yew
Dogwood	Cornus
Douglas_fir	Pseudotsuga
East_hophornbean	Ostrya virginiana
Elder	Sambucus
Elm	Ulmus
Eucalyptus	Eucalyptus spp
Fir_balsam	Abies balsamea
Fir_CA_red	Abies magnifica
Fir_corkbark	Abies lasiocarpa
Fir_fraser	Abies fraseri
Fir_grand	Abies grandis
Fir_noble	Abies procera
Fir_Pacf_silver	Abies amabilis
Fir_SantaLucia	Abies bracteata
Fir_Shasta_red	Abies magnifica shastensis
Fir_spp	Abies spp
Fir_subalpine	Abies lasiocarpa arizonica
Fir_white	Abies concolor

Land use category	Description
Gleditsia_locust	Waterlocust and Honeylocust
Hackberry	Celtis
Hawthorn	Crataegus
Hemlock	Tsuga
Hickory	Carya - includes pecan
Holly_American	Ilex
Hornbeam	Carpinus caroliniana
Incense_cedar	Calocedrus decurrens
Juniper	Juniperus - include Sou/Eas red cedar
KY_coffeetree	Gymnocladus dioicus
Larch	Larix
Loblolly_bay	Gordonia lasianthus
Madrone	Arbutus
Magnolia	Magnolia
Mahogany	Cercocarpus
Maple_bigleaf	Acer grandidentatum
Maple_bigtooth	Acer macrophyllum
Maple_black	Acer nigrum
Maple_boxelder	Acer negundo
Maple_FL	Acer barbatum
Maple_mtn	Acer spicatum
Maple_Norway	Acer platanoides
Maple_red	Acer rubrum
Maple_RkyMtn	Acer glabrum
Maple_silver	Acer saccharinum
Maple_spp	Acer spp
Maple_striped	Acer pensylvanicum
Maple_sugar	Acer saccharum
Mesquite	Prosopis
Misc_hardwoods	Misc hardwoods - undefined from FIA
Mixed_conifer_sp	Mixed conifers - underfined from FIA
Mountain_ash	Sorbus
Mulberry	Morus
Nyssa	Tupelo and blackgum
Oak_AZ_white	Quercus arizonica
Oak_bear	Quercus ilicifolia
Oak_black	Quercus velutina
Oak_blackjack	Quercus marilandica

Land use category	Description
Oak_blue	Quercus douglasii
Oak_bluejack	Quercus incana
Oak_bur	Quercus macrocarpa
Oak_CA_black	Quercus kelloggii
Oak_CA_live	Quercus agrifolia
Oak_CA_white	Quercus lobata
Oak_canyon_live	Quercus chrysolepis
Oak_chestnut	Quercus prinus
Oak_chinkapin	Quercus muehlenbergii
Oak_delta_post	Quercus stellata
Oak_Durand	Quercus durandii
Oak_Emery	Quercus emoryi
Oak_Engelmann	Quercus engelmannii
Oak_evergreen_sp	Quercus spp
Oak_Gambel	Quercus gambelii
Oak_interio_live	Quercus wislizenni
Oak_laurel	Quercus laurifolia
Oak_live	Quercus virginiana
Oak_Mexicanblue	Quercus oblongifolia
Oak_Northrn_pin	Quercus ellipsoidalis
Oak_Northrn_red	Quercus rubra
Oak_nuttall	Quercus nuttallii
Oak_OR_white	Quercus garryana
Oak_overcup	Quercus lyrata
Oak_pin	Quercus palustris
Oak_post	Quercus stellata
Oak_scarlet	Quercus coccinea
Oak_scrub	Quercus spp
Oak_shingle	Quercus imbricaria
Oak_Shumrd_red	Quercus shumardii
Oak_silverleaf	Quercus hypoleucoides
Oak_Southrn_red	Quercus falcata
Oak_spp	Quercus spp
Oak_swamp_cnut	Quercus michauxii
Oak_swamp_red	Quercus falcata
Oak_swamp_white	Quercus bicolor
Oak_turkey	Quercus laevis
Oak_water	Quercus nigra

Land use category	Description
Oak_white	Quercus alba
Oak_willow	Quercus phellos
Osage_orange	Maclura pomifera
Paulownia	Paulownia tomentosa
Pawpaw	Asimina triloba
Persimmon	Diospyros virginiana
Pine_Apache	Pinus engelmannii
Pine_Austrian	Pinus nigra
Pine_AZ	Pinus ponderosa
Pine_Bishop	Pinus muricata
Pine_blackjack	Pinus ponderosa
Pine_brstlcone	Pinus aristata
Pine_chihuahua	Pinus leiophylla
Pine_Coulter	Pinus coulteri
Pine_digger	Pinus sabiniana
Pine_Ewhite	Pinus strobus
Pine_foxtail	Pinus balfouriana
Pine_jack	Pinus banksiana
Pine_Jeffrey	Pinus jefferyi
Pine_knobcone	Pinus attenuata
Pine_limber	Pinus flexilis
Pine_loblolly	Pinus taeda
Pine_lodgepole	Pinus contorta
Pine_longleaf	Pinus palustris
Pine_Monterey	Pinus radiata
Pine_pinyon	Pinus monophylla
Pine_pinyon_brdr	Pinus discolor
Pine_pinyon_cmn	Pinus edulis
Pine_pitch	Pinus rigida
Pine_pond	Pinus serotina
Pine_ponderosa	Pinus ponderosa
Pine_red	Pinus resinosa
Pine_sand	Pinus clausa
Pine_scotch	Pinus sylvestris
Pine_shortleaf	Pinus echinata
Pine_slash	Pinus elliottii
Pine_spruce	Pinus glabra
Pine_sugar	Pinus lambertiana

Land use category	Description
Pine_Swwhite	Pinus strobiformis
Pine_tablemtn	Pinus pungens
Pine_VA	Pinus virginiana
Pine_Washoe	Pinus washoensis
Pine_whitebark	Pinus albicaulis
Pine_Wwhite	Pinus monticola
Pine_yellow	Pinus ponderosa
Populus	Cottonwood, poplars, and aspens
Prunus	Cherry and plum
Redbud	Cercis
Robinia_locust	Black and NM locust
Sassafras	Sassafras albidum
Sequoia	Redwood and giant sequoia
Serviceberry	Amelanchier arborea
Silverbell	Halesia
Smoketree	Cotinus obovatus
Soapberry_westrn	Sapindus drummondii
Sourwood	Oxydendrum arboretum
Sparkleberry	Vaccinium arboreum
Spruce_black	Picea marianna
Spruce_blue	Picea pungens
Spruce_Brewer	Picea breweriana
Spruce_Englemann	Picea engelmannii
Spruce_Norway	Picea abies
Spruce_red	Picea rubens
Spruce_Sitka	Picea sitchensis
Spruce_spp	Picea spp
Spruce_white	Picea glauca
Sweetgum	Liquidambar styraciflua
Sycamore	Platanus
Tallowtree_chins	Sapium sebiferum
Tamarix	Tamarix chinensis
Tanoak	Lithocarpus densiflorus
Torreya	Torreya californica
Tung_oil_tree	Aleurites fordii
Unknown_tree	Unknown spp
Walnut	Juglans - includes butternut
Water_elm	Planera aquatica

Land use category	Description
Willow	Salix
Yellow_poplar	Liriodendron tulipifera
Yellowwood	Cladrastis lutea
Yucca_Mojave	Yucca schidigera

8.9.2. Mobile

8.9.2.1. SMOKE-related Mobile Input Files

8.9.2.1.1. `CFPRO`: Mobile source control factor input file

The `CFPRO` file is used to allow the user to apply factors to **Movesmrg** to control hourly emission rates from all three MOVES lookup tables (i.e., RatePerDistance, RatePerVehicle, and RatePerProfile). Its format is shown in Table 8.57, "Format for `CFPRO`" [425].

A many-step hierarchy is used in applying the control factor cross-referencing. SMOKE preferentially matches each source to the most specific match in the file. The maximum field width in SMOKE and its input files for SCC is 20 characters as of SMOKE v4.0. The 8 or 10 digit SCC are still supported, but if the SCC is greater than 10 digits the SCC hierarchial approach will not be supported.

1. Country/state/county code, 10-digit SCC, month, and pollutant
2. Country/state/county code, 10-digit SCC, and month
3. Country/state/county code, 10-digit SCC, and pollutant
4. Country/state/county code, and 10-digit SCC
5. Country/state code, 10-digit SCC, month, and pollutant
6. Country/state code, 10-digit SCC, and month
7. Country/state code, 10-digit SCC, and pollutant
8. Country/state code, and 10-digit SCC
9. 10-digit SCC, month, and pollutant
10. 10-digit SCC, and month
11. 10-digit SCC, and pollutant
12. 10-digit SCC
13. Country/state/county code, month, and pollutant
14. Country/state/county code, and month
15. Country/state/county code, and pollutant
16. Country/state/county code
17. Country/state code, month, and pollutant
18. Country/statecode, and month
19. Country/state code, and pollutant
20. Country/state code
21. Ultimate default (no pollutant, no month, country/state/county code = 0, SCC = 0)

Table 8.57. Format for `CFPRO`

Column	Name	Type	Description
A	FIPS	Char(5)	FIPS code for state and county code
B	SCC	Char(20)	Source Category Code (SCC)
C	POL	Char(16)	Pollutant Code
D	Month	Integer	Processing month (Jan=1, Feb=2, and so on)
E	Control Factor	Real	Factor set by user to control hourly emission rates

Example 8.1. `CFPRO` file

```
13101,2201000330,CO,7,0.5
13101,2201000330,TOG,7,0.5
13102,2201000330,TOG,7,0.5
```

8.9.2.1.2. `MCXREF`: Mobile reference county cross-reference file

The `MCXREF` file is a list-directed file used to assign reference counties to inventory counties. Its format is shown in Table 8.58, "Format for `MCXREF`" [425].

Table 8.58. Format for `MCXREF`

Line	Position	Description
1+	A	One-digit country FIPS code for inventory county (Character)
	B	Two-digit state FIPS code for inventory county (Character)
	C	Three-digit county FIPS code for inventory county (Character)
	D	One-digit country FIPS code for reference county (Character)
	E	Two-digit state FIPS code for reference county (Character)
	F	Three-digit county FIPS code for reference county (Character)

Example 8.2. `MCXREF` file

```
0   13   101   0   13   121
0   13   102   0   13   121
0   13   103   0   13   121
0   13   121   0   13   121
0   13   123   0   13   217
0   13   125   0   13   217
0   13   127   0   13   217
0   13   015   0   13   217
0   13   017   0   13   217
0   13   019   0   13   217
. .  . .    . .   . .   . .   . .
```

8.9.2.1.3. `MEPROC`: List of mobile emission processes and associated activities and pollutants

The `MEPROC` file, used by **Spcmat** lists the pollutants that will be generated by MOVES for modeling on-roadway and off-network mobile sources. The available emission processes are listed in Section 2.8.4.9, " MOVES Emission Processes by Emissoin Rate Tables" [43]. The format of the `MEPROC` file is shown in Table 8.59, "Format for `MEPROC`" [426].

Table 8.59. Format for `MEPROC`

Line	Position	Description
1+	A	Activity name (Character)
	C	Associated pollutant name(s) for activity and process combination (Character)

Example 8.3. `MEPROC` example file for RatePerDistance MOVES table

```
VMT CO NO NO2 HONO SO2 NH3 TOG VOC_INV POC PEC PSO4 PNO3 PMFINE NH4 PMC
```

Example 8.4. `MEPROC` example file for RatePerVehicle MOVES table

```
VPOP CO NO NO2 HONO SO2 NH3 TOG VOC_INV POC PEC PSO4 PNO3 PMFINE NH4 PMC
```

8.9.2.1.4. `MFMREF`: Mobile fuel month file for MOVES

The `MFMREF` file specifies reference fuel months to assign to calendar months being simulated for each reference county. The file specifies the representative month where the fuel formulation mix is representative for other months. The file serves the purpose of grouping months of the year by fuel parameters. The format of the `MFMREF` file shown in Table 8.60, "Format for `MFMREF`" [426].

Table 8.60. Format for `MFMREF`

Line	Position	Description
1+	A	FIPS code for reference county (Char(5))
	B	Reference county fuel month (integer)
	C	Calendar month (integer)

Example 8.5. `MFMREF` file

```
013121  1   1
013121  1   2
013121  4   3
013121  4   4
013121  6   5
013121  6   6
```

8.9.2.1.5. `MRCLIST`: List of MOVES Lookup Tables

The `MRCLIST` file is an input file for **Movesmrg** that contains the list of MOVES-based lookup tables from the MOVES processing script associated with reference county and fuel month. The format of the `MRCLIST` file is shown in Table 8.61, "Format for `MRCLIST`" [427].

Table 8.61. Format for `MRCLIST`

Line	Position	Description
1+	A	FIPS code (Char(5))
	B	Fuel Month (Inetger)
	C+	Associated MOVES-based lookup table filename (Character)

Example 8.6. `MRCLIST` example file for Movesmrg (MOVES RPD lookup table)

```
13121 4 rateperdistance_13121_apr.txt
13121 7 rateperdistance_13121_jul.txt
```

Example 8.7. `MRCLIST` example file for Movesmrg (MOVES RPV lookup table)

```
13121 4 ratepervehicle_13121_apr.txt
13121 7 ratepervehicle_13121_jul.txt
```

Example 8.8. `MRCLIST` example file for Movesmrg (MOVES RPP lookup table)

```
13121 4 rateperprofile_13121_apr.txt
13121 7 rateperprofile_13121_jul.txt
```

8.9.2.1.6. `SPDPRO`: Mobile hourly speed profile input file

The `SPDPRO` file is used to provide weekend/weekday hourly speed profile data to **Movesmrg** to estimate hourly emission rates from RatePerDistance lookup table. It's format is shown in Table 8.62, "Format for `SPDPRO`" [428].

Table 8.62. Format for `SPDPRO`

Column	Name	Type	Description
1	FIPS	Char(5)	FIPS code for state and county code
2	SCC	Char(20)	Source Category Code (SCC)
3	Weekday	Char(7)	"WEEKDAY"
4	Hour 1	Real	Weekday hourly average speed (mile per hour) in local hour 1
5	Hour 2	Real	Weekday hourly average speed (mile per hour) in local hour 2
...	...	...	...
27	Hour 24	Real	Weekday hourly average speed (mile per hour) in local hour 24
28	Weekend	Char(7)	"WEEKEND"
29	Hour 1	Real	Weekend hourly average speed (mile per hour) in local hour 1
30	Hour 2	Real	Weekend hourly average speed (mile per hour) in local hour 2
...	...	...	...
52	Hour 24	Real	Weekend hourly average speed (mile per hour) in local hour 24

Example 8.9. `SPDPRO` **file**

```
13101,2201000330,WEEKDAY,35,,,,,,WEEKEND,34,,,,,,,,,
13102,2201000330,WEEKDAY,35,,,,,,WEEKEND,33,,,,,,,,,
13103,2201000330,WEEKDAY,36,,,,,,WEEKEND,38,,,,,,,,,
```

8.9.2.2. MOVES-related Mobile Input Files

8.9.2.2.1. Run Control File for MOVES Driver Script

The Run Control file (a.k.a Control.in) is used to specify the calendar year, pollutants, and day type (weekday or weekend); and the Met4moves output temperature file for the batch run of MOVES. Its format is shown in Table 8.63, "Format for `Control.in`" [429].

Table 8.63. Format for `Control.in`

Variable	Description
DBHOST	Database Hostname
MOVESHOME	Full pathname of MOVES installation directory
BATCHRUN	Character string that is the name for the MOVES batch files
OUTDIR	Full pathname where all output files will be written; batch files, MOVES runspec files, MOVES data importer files
MODELYEAR	Episode modeling year
POLLUTANTS	One or more keywords: PM, OZONE, TOXICS, GHG
DAYOFWEEK	Keywords WEEKDAY and/or WEEKEND. Optional input. If not specified, then both weekday and weekend estimates will be generated
METFILE	Full pathname of input meterological data (output from **Met4moves** processor).

Example 8.10. `Control.in` file

```
DBHOST          = localhost
MOVESHOME       = /netscr/MOVES2010
BATCHRUN        = GEORGIA
OUTDIR          = /netscr/MOVES2010/runspec_files/georgia
MODELYEAR       = 2005
POLLUTANTS      = ozone
DAYOFWEEK       = WEEKDAY, WEEKEND
METFILE         = /netscr/MOVESDATA/georgia/2005/2005_refcounty_MOVES_RH_OUTFILE
RPMETFILE       = /netscr/MOVESDATA/georgia/2005/2005_refcounty_MOVES_OUTFILE.tx
```

8.9.2.2.2. Reference County File for MOVES Driver Script

The Reference County file (a.k.a Countyrep.in) is used to specify the list of reference counties as separate REPCOUNTY packets, as shown below. Each reference county needs entries for the FIPS field and the full path and file names for csv files containing the MOVES data for AGE, IM, FUELSUPPLY, POP, Highway Performance Monitoring System (HPMS) and Vehicle Miles Travelled (VMT), and FUELFORM (optional). Every county specified in the Met4moves output file must be specified in the reference county file or an error will occur. The format is shown in Table 8.64, "Format for `Countyrep.in`" [430].

Table 8.64. Format for `Countyrep.in`

Variable	Description
FIPS	FIPS code for reference county (Char(5))
AGE	Full filename and pathname for age distribution csv file
IM	Full filename and pathname for the im coverage csv file
FUELSUPPLY	Full pathname and filename for the fuel supply csv file
POP	Full pathname and filename for the population csv file
HPMSVMT	Full pathname and filename for the HPMS Road Type and VMT csv file
FUELFORM (optional)	Full pathname and filename for the HPMS Road Type and VMT csv file

Example 8.11. `countyrep.in` file

```
<REPCOUNTY>
FIPS =13121
AGE  =/netscr/MOVES2010/runspec_files/georgia/agedistribution_13121.csv
IM   =/netscr/MOVES2010/runspec_files/georgia/imcoverage_13121.csv
FUELSUPPLY =/netscr/MOVES2010/runspec_files/georgia/fuelsupply_13121.csv
POP =/netscr/MOVES2010/runspec_files/georgia/population_13121.csv
HPMSVMT =/netscr/MOVES2010/runspec_files/georgia/hpmsvtypeyear_13121.csv
FUELFOR M =
<ENDREPCOUNTY>
<REPCOUNTY>
FIPS =13127
AGE =/netscr/MOVES2010/runspec_files/georgia/agedistribution_13127.csv
IM =/netscr/MOVES2010/runspec_files/georgia/imcoverage_13127.csv
FUELSUPPLY =/netscr/MOVES2010/runspec_files/georgia/fuelsupply_13127.csv
POP =/netscr/MOVES2010/runspec_files/georgia/population_13127.csv
HPMSVMT =/netscr/MOVES2010/runspec_files/georgia/hpmsvtypeyear_13127.csv
FUELFOR M =
<ENDREPCOUNTY>
```

8.9.2.2.3. Age Distribution File for Reference County for MOVES model

The Age Distribution file for each reference county shows the sourceTypeID, yearID, ageID, and ageFraction for the vehicles in each reference county. The format is shown in Table 8.65, "Format for agedistribution_referencecounty.csv" [430].

Table 8.65. Format for `agedistribution_referencecounty.csv`

Variable	Description
sourceTypeID	two-digit sourceTypeID for reference county (Integer)
yearID	four-digit year IDs
ageID	two-digit ageIDs
ageFraction	fraction of the vehicle population that is associated with the ageID for a specified year and sourceTypeID

Example 8.12. `agedistribution_13121.csv` file

```
sourceTypeID, yearID, ageID, ageFraction
11,2009,0,0.032258065
11,2009,1,0.032258065
11,2009,2,0.032258065
11,2009,3,0.032258065
11,2009,4,0.032258065
11,2009,5,0.032258065
11,2009,6,0.032258065
11,2009,7,0.032258065
11,2009,8,0.032258065
11,2009,9,0.032258065
11,2009,10,0.032258065
11,2009,11,0.032258065
11,2009,12,0.032258065
11,2009,13,0.032258065
11,2009,14,0.032258065
11,2009,15,0.032258065
11,2009,16,0.032258065
11,2009,17,0.032258065
11,2009,18,0.032258065
11,2009,19,0.032258065
11,2009,20,0.032258065
11,2009,21,0.032258065
11,2009,22,0.032258065
11,2009,23,0.032258065
11,2009,24,0.032258065
11,2009,25,0.032258065
11,2009,26,0.032258065
11,2009,27,0.032258065
11,2009,28,0.032258065
11,2009,29,0.032258065
11,2009,30,0.032258065
.....
```

8.9.2.2.4. Fuel Supply file for MOVES model

The Fuel Supply file is a list directed file used to define the fuel supply and the months that the fuels are used for each reference counties. Its format is shown in Table 8.66, "Format for `FuelSupply.csv`" [431].

Table 8.66. Format for `FuelSupply.csv`

Variable	Description
countyID	Country FIPS code for reference county (Char(5))
fuelYearID	Four-digit fuel year ID for reference county (Integer)
monthGroupID	One-digit month group ID for reference county (Integer)
fuelFormulationID	Four to Five digit fuel formulation ID for reference county (Integer)
marketShare	decimal fraction of marketShare for the fuel/month for the reference county
marketShareCV	decimal fraction (.5)?

Example 8.13. `fuelsupply_13121.csv` example file

```
"countyID","fuelYearID","monthGroupID","fuelFormulationID","marketShare","marketShare
13121,2009,1,8124,0.428571,0.5
13121,2009,1,2722,0.571429,0.5
13121,2009,1,20043,1.0,0.5
13121,2009,4,8136,0.428571,0.5
13121,2009,4,2734,0.571429,0.5
13121,2009,4,20043,1.0,0.5
```

8.9.2.2.5. Fuel Formulation file for MOVES model (optional)

The Fuel Formulation file is a list directed file used to specify a new fuelFormulation ID if a local fuel
does not match the list of nearly 9,000 predefined fuels in MOVES. Its format is shown in Table 8.66,
"Format for `FuelSupply.csv`" [431].

Table 8.67. Format for `FuelFormulation.csv`

Variable	Description
FuelFormulationID	Four to Five digit fuel formulation ID for reference county (Integer)
fuelSubtypeID	Four-digit fuel Subtype ID (Integer)
RVP	?-digit (Integer)
sulfurLevel	(Integer)
ETOHVolume	
MTBEVolume	
ETBEVolume	
TAMEVolume	
aromaticContent	
olefinContent	
benzeneContent	
e200	
e300	
bioDieselEsterVolume	
acetaneIndex	
PAHContent	

Example 8.14. `fuelformulation.csv` example file

```
"fuelFormulationID","fuelSubtypeID","RVP","sulfurLevel","ETOHVolume","MTBEVolume","TA
```

8.9.2.2.6. HPMSV Type for MOVES model

The HPMSV file is a list directed file used to define the annual VMT by Highway Performance Monitoring
System (HPMS) vehicle type. Its format is shown in Table 8.68, "Format for `HPMSV-
typeyear_countyrep.csv`" [433].

Table 8.68. Format for `HPMSVtypeyear_countyrep.csv`

Variable	Description
HPMSVtypeID	Two-digit HPMSV type IDs within the reference county (Integer)
yearID	year ID for the reference county (Integer)
HPMSVBaseYearVMT	
baseYearOffNetVMT	

Example 8.15. `hpmsvtypeyear_13121.csv` example file

```
HPMSVtypeID,yearID,HPMSBaseYearVMT,baseYearOffNetVMT
10,2009,100,0
20,2009,100,0
30,2009,100,0
40,2009,100,0
50,2009,100,0
60,2009,100,0
```

8.9.2.2.7. IM Coverage file for MOVES Model

The IM Coverage file is a list directed file used to define the inspection and maintanance data available for each reference counties. Its format is shown in Table 8.69, "Format for `imcoverage_reference-countyID.csv`" [433].

Table 8.69. Format for `imcoverage_referencecountyID.csv`

Variable	Description
polProcessID	Three-digit pol process in the reference county (Integer)
stateID	Two-digit state ID in the reference county (Integer)
countyID	Five-digit county ID for the reference county (Integer)
yearID	Four-digit year ID for reference county (Integer)
sourceTypeID	Two-digit source type ID for the reference county
fuelTypeID	One-digit fuel type ID
IMProgramID	One or two-digit inspection and maintance program ID for reference county (Integer)
inspectFreq	One-digit inspection frequency ID for the reference county
testStandardsID	Two-digit test standards ID for the reference county
begModelYearID	Four-digit beginning model year ID
endModelYearID	Four to Five digit fuel formulation ID for reference county (Integer)
useIMyn	Y - "does use I/M program", N - "does not use I/M program"
complianceFactor	decimal percentage compliance factor

Example 8.16. `imcoverage_13121.csv` example file

```
polProcessID,stateID,countyID,yearID,sourceTypeID,fuelTypeID,IMProgramID,inspectFreq,
113,13,13121,2009,31,1,8,1,43,1996,2005,Y,97
113,13,13121,2009,31,1,7,1,41,1975,1995,Y,97
113,13,13121,2009,21,1,8,1,43,1996,2005,Y,97
113,13,13121,2009,21,1,7,1,41,1975,1995,Y,97
301,13,13121,2009,31,1,10,1,51,1996,2005,Y,94.09
301,13,13121,2009,31,1,4,1,24,1975,1995,Y,94.09
301,13,13121,2009,21,1,10,1,51,1996,2005,Y,94.09
301,13,13121,2009,21,1,4,1,24,1975,1995,Y,94.09
.....
```

8.9.2.2.8. MOVES Vehicle Population by source type input file for MOVES model

The MOVES Population by source type file contains the year, source and population for each reference county. Its format is shown in Table 8.70, "Format for `population_13217.csv`" [434].

Table 8.70. Format for `population_13217.csv`

Variable	Description
yearID	Four-digit source type ID (Integer)
sourceTypeID	two-digit source type ID (Integer)
sourceTypePopulationID	Three-digit source type population ID (Integer)

Example 8.17. `population_13121.csv` example file

```
yearID,sourceTypeID,sourceTypePopulation
2009,11,100
2009,21,100
2009,31,100
2009,32,100
2009,41,100
2009,42,100
2009,43,100
2009,51,100
2009,52,100
2009,53,100
2009,54,100
2009,61,100
2009,62,100
```

8.9.2.2.9. Dummy Average Speed Distribution file for MOVES model

The Dummy Average Speed Distribution file is a file that contains the average speed bin ID and average speed fraction for each reference counties. Its format is shown in Table 8.71, "Format for `dummyavg-speeddistribution.csv`" [435].

Table 8.71. Format for `dummyavgspeeddistribution.csv`

Variable	Description
sourceTypeID	Two-digit source type ID (Integer)
roadTypeID	one-digit road type ID (Integer)
hourDayID	Two-digit hour day ID (Integer)
avgSpeedBinID	average speed bin ID (Integer)
avgSpeedFraction	decimal fraction representing the proportion of vechicles that travel at within each of the speed bins for a specific road typeID and hourDayID for the reference county

Example 8.18. `dummy_avgspeeddistribution.csv` example file

```
sourceTypeID,roadTypeID,hourDayID,avgSpeedBinID,avgSpeedFraction
11,2,12,1,0
11,2,12,2,0
11,2,12,3,0
11,2,12,4,0
11,2,12,5,0
11,2,12,6,0
11,2,12,7,0
11,2,12,8,0
11,2,12,9,0
11,2,12,10,0
11,2,12,11,0
11,2,12,12,0.0091073
11,2,12,13,0.0285067
11,2,12,14,0.0988925
11,2,12,15,0.248298
11,2,12,16,0.615196
..
```

8.9.2.2.10. Dummy Day VMT Fraction file for MOVES model

The Dummy Day VMT Fraction file is a file that contains specifies whether the VMT are weekday or weekend. Its format is shown in Table 8.72, "Format for dummydayvmtfraction.csv" [435].

Table 8.72. Format for `dummydayvmtfraction.csv`

Variable	Description
sourceTypeID	Two-digit source type ID (Integer)
monthID	one or two-digit month ID (Integer)
roadTypeID	one digit road type ID (Integer)
dayID	2 - weekday, 5 - weekend
dayVMTFraction	decimal fraction representing the proportion of vehicle miles traveled for the weekday or weekend day type

Example 8.19. `dummy_roadtypedistribution.csv` example file

```
sourceTypeID,monthID,roadTypeID,dayID,dayVMTFraction
11,7,1,2,0.237635
11,7,1,5,0.762365
11,7,2,2,0.27882
11,7,2,5,0.72118
11,7,3,2,0.27882
11,7,3,5,0.72118
11,7,4,2,0.237635
11,7,4,5,0.762365
11,7,5,2,0.237635
11,7,5,5,0.762365
21,7,1,2,0.237635
21,7,1,5,0.762365
21,7,2,2,0.27882
21,7,2,5,0.72118
21,7,3,2,0.27882
21,7,3,5,0.72118
21,7,4,2,0.237635
21,7,4,5,0.762365
21,7,5,2,0.237635
21,7,5,5,0.762365
..
```

8.9.2.2.11. Dummy Hour VMT Fraction file for MOVES model

The Dummy Hour VMT Fraction file is a file that contains specifies the VMT fraction associated with each hour of the day. Its format is shown in Table 8.73, "Format for `dummyhourvmtfraction.csv`" [436].

Table 8.73. Format for `dummyhourvmtfraction.csv`

Variable	Description
sourceTypeID	Two-digit source type ID (Integer)
roadTypeID	one digit road type ID (Integer)
dayID	one digit day type ID (Integer), 2 - weekday, 5 - weekend
hourID	1-24 hours
hourVMTFraction	decimal fraction representing the proportion of vehicle miles traveled are associated with a particular hour of the day for each source type, road type and day type (weekday or weekend)

Example 8.20. `dummy_hourvmtfraction.csv` example file

```
sourceTypeID,roadTypeID,dayID,hourID,hourVMTFraction
11,1,2,1,0.0214739
11,1,2,2,0.0144428
11,1,2,3,0.0109684
11,1,2,4,0.00749451
11,1,2,5,0.00683855
11,1,2,6,0.0103588
11,1,2,7,0.0184304
11,1,2,8,0.0268117
11,1,2,9,0.0363852
11,1,2,10,0.0475407
11,1,2,11,0.0574664
11,1,2,12,0.0650786
11,1,2,13,0.0713228
11,1,2,14,0.0714917
11,1,2,15,0.0717226
11,1,2,16,0.0720061
11,1,2,17,0.0711487
11,1,2,18,0.0678874
11,1,2,19,0.0617718
11,1,2,20,0.0516882
11,1,2,21,0.0428658
11,1,2,22,0.0380302
11,1,2,23,0.0322072
11,1,2,24,0.0245677
..
```

8.9.2.2.12. Dummy Month VMT Fraction file for MOVES model

The Dummy Month VMT Fraction file is a file that contains specifies the VMT fraction associated with each hour of the day. Its format is shown in Table 8.74, "Format for `dummymonthvmtfraction.csv`" [437].

Table 8.74. Format for `dummymonthvmtfraction.csv`

Variable	Description
sourceTypeID	Two-digit source type ID (Integer)
isLeapYear	Y or N
monthID	one digit day type ID (Integer), 2 - weekday, 5 - weekend
monthVMTFraction	decimal fraction representing the proportion of vehicle miles traveled are associated with a specific month for each source type.

Example 8.21. `dummy_monthvmtfraction.csv` example file

```
sourceTypeID,isLeapYear,monthID,monthVMTFraction
11,N,1,0.083333333
11,N,2,0.083333333
11,N,3,0.083333333
11,N,4,0.083333333
11,N,5,0.083333333
11,N,6,0.083333333
11,N,7,0.083333333
11,N,8,0.083333333
11,N,9,0.083333333
11,N,10,0.083333333
11,N,11,0.083333333
11,N,12,0.083333333
21,N,1,0.083333333
21,N,2,0.083333333
21,N,3,0.083333333
21,N,4,0.083333333
21,N,5,0.083333333
21,N,6,0.083333333
21,N,7,0.083333333
21,N,8,0.083333333
21,N,9,0.083333333
21,N,10,0.083333333
21,N,11,0.083333333
21,N,12,0.083333333
..
```

8.9.2.2.13. Dummy Road Type Distribution file for MOVES model

The Road Type Distribution file is a file that contains the average speed bin ID and average speed fraction. Its format is shown in Table 8.75, "Format for `dummyroadtypedistribution.csv`" [438].

Table 8.75. Format for `dummyroadtypedistribution.csv`

Variable	Description
sourceTypeID	Two-digit source type ID (Integer)
roadTypeID	one-digit road type ID (Integer)
roadTypeVMTFraction	decimal fraction representing the proportion of vehicle miles traveled that occur on a specific road type

Example 8.22. `dummy_roadtypedistribution.csv` example file

```
sourceTypeID,roadTypeID,roadTypeVMTFraction
11,1,0.2
11,2,0.2
11,3,0.2
11,4,0.2
11,5,0.2
21,1,0.2
21,2,0.2
21,3,0.2
21,4,0.2
21,5,0.2
..
```

8.9.3. Point

The directories where the point-specific input files should be stored are listed in Table 8.10, "Point-Specific Input File Locations" [350]. The table also lists the programs that use each input file.

8.9.3.1. `PELVCONFIG`: Elevated source selection file

The `PELVCONFIG` is optionally used by the **Elevpoint** program. It permits users to provide SMOKE criteria for grouping stacks and for selecting elevated and PinG point sources. "Stack groups" are groups of stacks from the same facility that are treated by SMOKE and the AQM as a single stack when computing plume rise and/or processing the stack as a PinG source. They have the effect of reducing the number of elevated and PinG stacks which makes for faster processing, particularly in the case of PinG sources in the AQM. Users can specify stack groups regardless of whether elevated or PinG sources are also being specified.

All emissions in a stack group are summed for the purposes of comparing against the elevated source criteria and PinG source criteria. In short, the grouped sources are treated as a single source for all SMOKE features regarding elevated source selection and special outputs. The grouped sources can still have different temporal profiles and chemical speciation profiles. For more information on how the emissions are computed for source selection, see Section 6.3, "**Elevpoint**" [223].

The `PELVCONFIG` file has four sections: (1) source category selection, (2) source grouping, (3) PinG source selection, and (4) elevated source selection. Sections 2, 3, and 4 are optional, but **Elevpoint** will not run if all three sections are missing, because in that case the file would be instructing it to do nothing. The sections need not appear in any particular order, but each section is permitted only once. Comments may also be used in the file. The following subsections explain these major input features.

8.9.3.1.1. Using comments

Comments can be put in the file using a pound sign (#) in the first column of a line or a double pound sign (##) anywhere else in a line. For example, the double pound can appear after packet labels or instructions in the file.

8.9.3.1.2. Source category selection

This section is used to tell **Elevpoint** which source category to process. Since **Elevpoint** is designed only for point sources, this is very simple. This section needs to appear as follows:

SMK_SOURCE P

8.9.3.1.3. Source grouping

The /SPECIFY ELEV GROUPS/ packet is used for defining source groups based on source criteria. The packet contains the following entries:

- /SPECIFY ELEV GROUPS/

  Must appear to start the packet. Only one packet of this type is permitted.

- <Source grouping criteria, explained next>

- /END/

  Must appear to end the packet.

The *source grouping criteria* can be as many lines as you want. Each line contains at least three parts: (1) variable (explained in Section 8.9.3.1.3.1, "/SPECIFY ELEV GROUPS/ variables" [440]), (2) type of criterion (explained in Section 8.9.3.1.3.2, "/SPECIFIC ELEV GROUPS/ grouping criteria" [441]), and (3) value for criterion, in the following structure:

<variable> <type of criterion> <value for criterion>

For example, the following source grouping criterion groups all inventory sources at the same facility that have stack heights within ±2 meters of weighted average stack height:

```
HT  +/-  2.0
```

The fact that sources are grouped by facility is implicit to the **Elevpoint** program, since we assumed when creating **Elevpoint** that grouping at a coarser level of detail was not desirable. The packet supports boolean OR and AND conditions. Each separate line of the packet is an OR condition. The AND conditions must appear on a single line. Any number of lines are permitted in this packet, as are any number of AND conditions or AND/OR combinations. However, each line has a maximum length of 300 characters.

The following example will help illustrate. To indicate that one group should consist of sources with stack heights within ±2 meters of weighted average stack height that also have diameters within ±1 meter of weighted average diameter, and that a second group should consist of sources with exit velocities below 10 m/s, the packet would be:

```
/SPECIFY ELEV GROUPS/
HT  +/-  2.0  AND  DIAMETER  +/- 1.0
VE  <  10.
/END/
```

8.9.3.1.3.1. /SPECIFY ELEV GROUPS/ variables

The elevated groups packet supports the following variables:

- HT: stack height (m)

- DM: stack diameter (m)

- TK: stack gas exit temperature (K)

- VE: stack gas exit velocity (m/s)

- FL: stack gas exit flow rate (m^3/s)

- <pollutant name>: whatever pollutant names are in the inventory (e.g., NOX). We do not know why a user might want this type of grouping, but SMOKE does not prevent it.

8.9.3.1.3.2. /SPECIFIC ELEV GROUPS/ grouping criteria

The elevated groups packet supports the following types of grouping criteria:

- +/- or -/+: variable values must be within plus or minus <value for criterion> for the variable given

- >: variable values must be greater than <value for criterion> for the variable given

- >= or =>: variable values must be greater than or equal to <value for criterion> for the variable given

- <: variable values must be less than <value for criterion> for the variable given

- <= or =<: variable values must be less than or equal to <value for criterion> for the variable given

- = or ==: variable values must be equal to <value for criterion> for the variable given

- %: variable values must be less than plus or minus <value for criterion> percent different for the variable given

8.9.3.1.4. Plume-in-grid source selection

The /SPECIFY PING/ packet is used for selecting plume-in-grid sources based on source criteria. The packet contains the following entries:

- /SPECIFY PING/

  Must appear to start the packet. Only one packet of this type is permitted.

- <Source selection criteria, explained next>

- /END/

  Must appear to end the packet.

The *source selection criteria* can be as many lines as you want. Each line contains at least three parts: (1) variable (explained in Section 8.9.3.1.4.1, "/SPECIFY PING/ variables" [441]), (2) type of criterion (explained in Section 8.9.3.1.4.2, "/SPECIFY PING/ selection criteria" [442]), and (3) value for criterion in the following structure:

<variable> <type of criterion> <value for criterion>

This structure is the same as that of the /SPECIFY ELEV GROUPS/ packet, and the description of the PinG source selection criteria is the same as that of the source grouping criteria (see Section 8.9.3.1.3, "Source grouping" [440]).

8.9.3.1.4.1. /SPECIFY PING/ variables

The PinG packet supports the same variables as the /SPECIFY ELEV GROUPS/ packet (see Section 8.9.3.1.3.1, "/SPECIFY ELEV GROUPS/ variables" [440]) and the following additional variables:

- RISE: The plume rise as computed by the Briggs algorithm documented with the **Elevpoint** program (see Section 6.3, "**Elevpoint**" [223]). The <value for criterion> must be provided in meters. To select all sources with plume rise greater than or equal to 75 meters, you would input:

```
RISE >= 75.
```

- SOURCE: Source number from the intermediate inventory file output by **Smkinven**. This variable permits selection of specific SMOKE sources as PinG sources.

- PLANT: Plant ID code from the intermediate inventory file output by **Smkinven**. The "PLANT" variable can only be paired with the "IS" criterion type (below); no other criterion types (such as "<" or "=") are permitted. It is recommended that this variable be used in addition to the FIPS variable (below) to allow selection of all records at specific plants. It can also be combined with stack parameter variables.

- FIPS: Country/state/county region code. This can be used with the "PLANT" variable or other variables to specific PinG sources by region. It is not recommended that this variable be used alone for specifying PinG sources, because that would result in insignificant sources being prepared for the AQM as PinG sources. It does not make sense to process such sources as PinG sources in AQMs.

8.9.3.1.4.2. /SPECIFY PING/ selection criteria

The PinG packet supports the same types of criteria as the /SPECIFY ELEV GROUPS/ packet (see Section 8.9.3.1.3.2, "/SPECIFIC ELEV GROUPS/ grouping criteria" [441]) and the following additional criteria types:

- TOP: This criterion type permits users to select sources with the largest values of the variable given. The <value for criterion> is the number of sources that will be selected based on the value given. This criterion type is permitted only when using a pollutant name as the <variable>.

- IS: This criterion type must be used to assign a value to the "PLANT" variable.

The following example illustrates a PinG packet that uses several of the variables and criterion types. In the following section is a further example that demonstrates using the rest of the variables and criterion types.

```
/SPECIFY PING/
RISE >= 150.
HT >= 100.
NOX TOP 100 AND SO2 TOP 100
NOX > 500.
/END/
```

8.9.3.1.5. Elevated source selection

The /SPECIFY ELEV/ packet is used for selecting elevated sources based on source criteria. The packet contains the following entries:

- /SPECIFY ELEV/

  Must appear to start the packet. Only one packet of this type is permitted.

- <Source selection criteria, explained above>

- /END/

  Must appear to end the packet.

All of the source selection criteria for elevated sources are the same as the PinG selection criteria at this time (see Section 8.9.3.1.4, "Plume-in-grid source selection" [441]). To use the "cutoff" method for UAM-style processing, only the "RISE" variable should be used; the packet should not contain any other selection criteria. The "cutoff" method is the traditional approach for elevated-source selection for UAM, UAM-V, REMSAD, and CAM$_X$; you can decide if you want to use this traditional approach or a more complex set of criteria.

The following example illustrates using all of the different source selection variables and criterion types listed in the previous sections.

```
/SPECIFY ELEV/
RISE >= 40. AND RISE < 150.
NOX TOP 500
NH3 TOP 50
FIPS > 37001 AND FIPS <= 37031
SOURCE > 100 AND SOURCE < 200
/END/
```

8.9.3.2. `PSTK`: Point source stack replacement parameters file

The `PSTK` file is an ASCII file that contains the replacement stack parameters that **Smkinven** uses when stack parameters are missing from the inventory. The format of the file permits stack parameters to be replaced based on the point sources' country/state/county code and SCC and is shown in Table 8.76, "Format for `PSTK`" [443].

Table 8.76. Format for `PSTK`

Line	Position	Description
1+	A	Country/State/County code (Character)
	B	Twenty-digit SCC (Character)
	C	Stack height from ground (m) (Real)
	D	Stack diameter (m) (Real)
	E	Stack exhaust temperature (K) (Real)
	F	Stack exhaust velocity (m/s) (Real)

8.9.3.3. `PTMPLIST`: List of hourly point source emissions files

The `PTMPLIST` file is an list file containing the full name and path of available hourly point source emissions files. This file is created automatically from the example SMOKE run scripts. Its format is shown in Table 8.77, "Format for `PTMPLIST`" [443].

Table 8.77. Format for `PTMPLIST`

Line	Position	Description
1+	A	Full name and path of hourly point source emissions file

8.10. Other Support Files

The directories where the other support files should be stored are listed in Table 8.11, "Other Support Input File Locations" [350]. The table also lists the programs that use each input file.

8.10.1. `ARTOPNT`: Area-to-point conversions file

The `ARTOPNT` file provides the point-source locations to assign to stationary area and nonroad mobile sources. The file is separated into sections. Each section starts with a header that contains the SCCs that will receive the point-source locations and allocation factors contained in that section. Within the section, the content and format are similar to the EMS-HAP airport allocation factor files (e.g., `ap_afl.sas7bdat`).

Each section of the file starts with the following packet:

/LOCATIONS/ <SCC 1> <SCC 2> ... <SCC 9>

for SCCs 1 through 9 that apply to the section. The SCCs must be full 10-digit SCCs separated by spaces. A maximum of 9 SCCs may appear in one /LOCATIONS/ packet. If more than 9 SCCs are needed for a specific section, then multiple /LOCATIONS/ packets can be included in subsequent rows prior to the section to which they apply. In other words, two or more lines in sequence may include the/LOCATIONS/ packet, and all of the SCCs from both packets will be applied to the subsequent section of the file.

Within each section delineated by the /LOCATIONS/ packets, the format is shown in Table 8.78, "Format for `ARTOPNT`" [444].

Table 8.78. Format for `ARTOPNT`

Position	Name	Type	Description
1-6	COID	Int	Country, state, and county code (CSSYYY, where C = country, SS=state, and YYY=county) (required)
8-9	STABBR	Char 2	Postal abbreviation for state (not used by SMOKE)
11-32	COUNTY	Char 22	County name (not used by SMOKE)
34-37	LOCID	Char 4	Airport ID (unused for now, required for ISCST3)
39-61	NAME	Char 23	Airport name (if name contains spaces, commas, or semi-colons, then the entire name must be enclosed in double-quotes)
63-84	CITY	Char 22	City name (not used by SMOKE)
86-93	ARPT_USE	Char 8	Airport purpose (not used by SMOKE)
97-109	LON	Real	Longitude for the emissions (required)
114-124	LAT	Real	Latitude for the emissions (required)
126-139	ALLOC	Real	Allocation factor for this lat/lon, county, and SCC (required)

The section ends with the start of another packet (indicated by a new /LOCATIONS/ header line or with the end of the file.

8.10.2. `COSTCY`: **Country, state, and county names and data**

The COSTCY file is used to assign numeric codes to the various countries, states, and counties to be modeled; the file also provides information concerning default time zones and Daylight Savings observance. A header line can be used in this file to indicate that population data are contained in the file as the last column on each line of the /COUNTY/ packet. The header line and additional population column are optional. If it is present in the file, it must be on the first line of the file and use the following syntax:

#POPULATION <year>

where <year> is a 4-digit year.

The country, state, and county codes in each section of this file must be sorted in increasing order. If they are not, the programs reading this file will give an error. The format of the COSTCY file is shown in Table 8.79, "Format for COSTCY" [446].

Table 8.79. Format for COSTCY

Line	Position	Description
1	A	Start of country section using packet /COUNTRY/
2+	1	Country code (Character)
	3-22	Country name (Character)
3	A	Start of state section using packet /STATE/
4+	1	Country code; should match an entry in the /COUNTRY/ section (Character)
	2-3	State code (Character)
	4-5	State abbreviation (Character) (not used by SMOKE)
	7-26	State name (Character)
	27-28	EPA Region (Integer) (not used by SMOKE)
	32-34	Standard (non-Daylight-Saving) state time zone; see Table 8.80, "Valid COSTCY and GEOCODE_LEVEL4 time zones" [447] for valid time zones (Character) (optional)
5	A	Start of county section using packet /COUNTY/
6+	2-3	State abbreviation, should match an entry in the /STATE/ section (Character) (not used by SMOKE)
	5-24	County name (Character)
	26	Country code, should match the one in the /COUNTRY/ section (Character)
	27-28	State code, should match the one in the /STATE/ section (Character)
	29-31	County code (Character)
	32-34	AEROS State Code (Integer) (not used by SMOKE)
	35-38	AEROS County Code (Integer) (not used by SMOKE)
	40-42	Standard (non-Daylight-Saving) county time zone; see Table 8.80, "Valid COSTCY and GEOCODE_LEVEL4 time zones" [447] for valid time zones (Character)
	43	Leaving this field blank indicates that Daylight Saving time is used in the county. Any character in this field will turn off the use of Daylight Saving time in the county.
	44-52	County center longitude (degrees) (Real) (not used by SMOKE)
	53-61	County center latitude (degrees) (Real) (not used by SMOKE)
	63-74	County area (square miles) (Real) (not used by SMOKE)
	76-84	Westernmost longitude (degrees) (Real) (not used by SMOKE)
	86-94	Easternmost longitude (degrees) (Real) (not used by SMOKE)
	95-103	Southernmost latitude (degrees) (Real) (not used by SMOKE)
	104-112	Northernmost latitude (degrees) (Real) (not used by SMOKE)
	114-128	Population for the county (optional; used only by **Smkreport** with the NORMALIZE POPULATION instruction)

Table 8.80. Valid `COSTCY` and `GEOCODE_LEVEL4` time zones

GMT	Time zone	Description
+12:00	BIT	Baker Island Time
+11:00	SST	Samoa Standard Time
+10:00	HST	Hawaii Standard Time
+9:00	AKT	Alaska Standard Time
+8:00	PST	Pacific Standard Time
+7:00	MST	Mountain Standard Time
+6:00	CST	Central Standard Time
+5:00	EST	Eastern Standard Time
+4:00	AST	Atlantic Standard Time
+3:00	ART	Argentina Time
+2:00	FNT	Fernando de Noronha Time
+1:00	EGT	Eastern Greenland Time
0:00	GMT	Greenwich Mean Time
-1:00	CET	Central European Time
-2:00	EET	Eastern European Time
-3:00	MSK	Moscow Time
-4:00	GST	Gulf Standard Time
-5:00	PKT	Pakistan Standard Time
-6:00	BST	Bangladesh Standard Time
-7:00	THA	Thailand Time
-8:00	HKT	China Standard Time
-9:00	KST	Korean Standard Time
-10:00	AET	Austrailian Eastern Time
-11:00	ADT	Austrailian Daylight Savings Time
-12:00	FJT	Fiji Time
-13:00	NZT	New Zealand Daylight Time
-14:00	LNT	Line Islands Time

8.10.3. `GEOCODE_LEVEL[1-4]: Geographical Input Files`

There are four GEOCODE_LEVEL files that are used to specify country level (Section 8.10.3.1, "`GEO-CODE_LEVEL1`" [448]), country/state-level (Section 8.10.3.2, "`GEOCODE_LEVEL2`" [448]), country/state/county-level (Section 8.10.3.3, "`GEOCODE_LEVEL3`" [449]) or country/state/county/tribal-level (Section 8.10.3.4, "`GEOCODE_LEVEL4`" [449]) Geographical codes and descriptions. These files are used if USE_EXP_GEO_CODES is set to Y, and they replace the (Section 8.10.2, "`COSTCY`: Country, state, and county names and data" [445]) file.

8.10.3.1. `GEOCODE_LEVEL1`

The GEOCODE_LEVEL1 input file contains a 3 character code used to specify the Country-level Geographical Code. The COSTCY file will not be used if the environment variable USE_EXP_GEO_CODES is set to Y. If the user would like to use chemical and temporal profiles that are specified for each country in the gridded inventory data then they would create a GEOCODE_LEVEL1 file.

The GEOCODE_LEVEL1 file is comma delimited as shown in Table 8.81, "File Format for GEO-CODE_LEVEL1" [448] . This file should not include any descriptions that have the pipe symbol (|), single quotes, or double quotes. It is okay to include dashes, slashes, semicolons, colons, commas, and periods in descriptions. The format of the GEOCODE_LEVEL1 file is shown in Table 8.81, "File Format for GEOCODE_LEVEL1" [448].

```
#GEOCODE_LEVEL1, Description
"001","United States of America"
"002","Mexico"
"003","South Korea"
"004","Ocean"
```

Table 8.81. File Format for `GEOCODE_LEVEL1`

Lines	Position	Description
1+	A	Country Code(3-character maximum string)
	B	Country Code Description (20-character string)

8.10.3.2. `GEOCODE_LEVEL2`

The GEOCODE_LEVEL2 input file contains up to 6 character code to specify the Country/State-level Geographical Codes which will replace the COSTCY file if the environment variable USE_EXP_GEO_CODES is set to Y. If the user would like to use chemical and temporal profiles that are specified for each state level region for each country in the gridded inventory data then they would create a GEOCODE_LEVEL2 file.

The GEOCODE_LEVEL2 file is comma delimited as shown in Table 8.82, "File Format for GEO-CODE_LEVEL2" [449] . This file should not include any descriptions that have the pipe symbol (|), single quotes, or double quotes. It is okay to include dashes, slashes, semicolons, colons, commas, and periods in descriptions. The format of the GEOCODE_LEVEL2 file is shown in Table 8.82, "File Format for GEOCODE_LEVEL2" [449].

```
#GEOCODE_LEVEL2, Description
"001006","California, USA"
"001035","New Mexico, USA"
"001048","Texas, USA"
"002030","Veracruz, Mexico"
```

Table 8.82. File Format for `GEOCODE_LEVEL2`

Lines	Position	Description
1+	A	Country Code(6-character maximum string)
	B	Country Code Description (20-character string)

8.10.3.3. `GEOCODE_LEVEL3`

The GEOCODE_LEVEL3 input file contains up to 9 character code to specify the Country/State-level/County-level Geographical Code which will replace the COSTCY file if the environment variable USE_EXP_GEO_CODES is set to Y.

The `GEOCODE_LEVEL3` file is comma delimited as shown in Table 8.83, "File Format for `GEO-CODE_LEVEL3`" [449] . This file should not include any descriptions that have the pipe symbol (|), single quotes, or double quotes. It is okay to include dashes, slashes, semicolons, colons, commas, and periods in descriptions. The format of the `GEOCODE_LEVEL3` file is shown in Table 8.83, "File Format for `GEOCODE_LEVEL3`" [449].

```
#GEOCODE_LEVEL3, Description
"001006037","Los Angeles, California, USA"
"001035028","Los Alamos, New Mexico, USA"
"001048015","Austin, Texas, USA"
```

Table 8.83. File Format for `GEOCODE_LEVEL3`

Lines	Position	Description
1+	A	Country Code(9-character maximum string)
	B	Country Code Description (20-character string)

8.10.3.4. `GEOCODE_LEVEL4`

The GEOCODE_LEVEL4 input file contains a 12 character code to specify the Country/State/County/Tribal-level Geographical Code which will replace the COSTCY file if the environment variable USE_EXP_GEO_CODES is set to Y.

The `GEOCODE_LEVEL4` file is comma delimited as shown in Table 8.84, "File Format for `GEO-CODE_LEVEL4`" [450] . This file should not include any descriptions that have the pipe symbol (|), single quotes, or double quotes. It is okay to include dashes, slashes, semicolons, colons, commas, and periods in descriptions. The format of the `GEOCODE_LEVEL4` file is shown in Table 8.84, "File Format for `GEOCODE_LEVEL4`" [450].

```
#GEOCODE_LEVEL4, Description, Time Zone
"001006037000","Los Angeles, CA, USA",PST
"001035028000","Los Alamos, NM, USA",CST
"001048015000","Austin, TX, USA",CST
```

Table 8.84. File Format for `GEOCODE_LEVEL4`

Lines	Position	Description
1+	A	Country Code(12-character maximum)
	B	Country Code Description (20-character)
	C	Time Zone (3-character)
	D	Standard (non-Daylight-Saving) county time zone; see Table 8.80, "Valid `COSTCY` and `GEOCODE_LEVEL4` time zones" [447] for valid time zones (Character)

8.10.4. `INVTABLE`: Inventory table

The `INVTABLE` file replaces the `SIPOLS` and `ACTVNAMS` files previously used by SMOKE. **Smkinven** uses the inventory table to select the inventory pollutants that will be output by **Smkinven**. The valid pollutants are again checked at the end of SMOKE processing by **Smkmerge**, to ensure that the program knows about all of the pollutants and also to set the order of the output variables (output species are sorted by the order of the pollutants in the inventory table and for multiple species per pollutant, by alphabetical order). You can use the `INVTABLE` to drop inventory pollutants in **Smkinven** without having to modify the inventory files, simply by modifying `INVTABLE`. The units used in the SMOKE intermediate files are listed in this file. The SMOKE programs **Spcmat**, and **Movesmrg** use this file to determine which pollutants should be included in the NONHAPVOC or NONHAPTOG calculation.

The inventory table also provides a way to assign the SPECIATE4 species ID to any pollutant that is a Hazardous Air Pollutant (HAP) VOC. This assignment is not used directly by SMOKE, but rather is used by the Speciation Tool when creating VOC speciation profiles. This is used only when creating speciation profiles to support SMOKE runs that will combine criteria VOC and HAP VOC emissions, using the `NHAPEXCLUDE` file in the **Smkinven** program. More specifically, this can be done for the purpose of using VOC HAP emissions to augment VOC speciation profiles. Since the Inventory Table provides information to the Speciation Tool, this field is not used by SMOKE, but is included for continued integration of SMOKE inputs with the Speciation Tool.

As described in the **Smkinven** documentation in Chapter 6, **Smkinven** in some cases will create "no-integrate" (NOI) HAP pollutant emissions from the inventory data by renaming the pollutant with a "_NOI" suffix applied to the end of the Inventory Data Name from the Inventory Table for such pollutants. Similarly, **Smkinven** will automatically add all possibly needed "_NOI" pollutant names to the internal master list of valid pollutant names. This list would otherwise be limited to the contents of the `INVTABLE` file entries where the Keep column contains an "Y". **Smkinven** inserts these names in the master list such that the "_NOI" version of the pollutant name appears in the list just after the polutant on which the name is based. The inserted entries that include "_NOI" are made when the following values are included for the pollutant in the `INVTABLE` file.

• A "Y" appears in the "Keep" field

• A "V" or "T" appears in the "VOC or TOG component" field

• A "Y" appears in the "Model species" field

• Something other than "Y" appears in the "Explicit in mechanism" field (e.g., "N" or blank)

It is possible for a user to choose two pollutants that result in double counting the emissions from the inventory. For example, you may want to model both benzo(a)pyrene and 7-PAH which, would result in the emissions associated with CAS number 205992 (for benzo(a)pyrene) to be duplicated. The user is responsible for the choices made and the resulting changes in inventory mass.

The format of the INVTABLE is shown in Table 8.85, "Format for INVTABLE" [451].

Table 8.85. Format for INVTABLE

Position	Description	Type
1-11	Inventory data name	11-character string
13-15	Mode (optional) e.g., EXR, EXS, EXH, EVP, BIO	3-character string
17-32	Inventory Pollutant Code	16-character string
34-38	SPECIATE4 ID(optional)	Integer
40	Reactivity group (not used by SMOKE)	Integer
42	Keep (Y or N)	1-character string
44-49	Factor	Float
50	VOC or TOG component (V/T/N)	1-character string
52	Model species (Y/N)	1-character string
54	Explicit in mechanism (Y/N)	1-character string
56	Activity	1-character string
58-60	NTI	Integer
62-77	Units to be used in SMOKE intermediate files (*changing not recommended*)	16-character string
79-118	Inventory data description - a description of the name provided in columns 1-11.	40-character string
119-158	CAS pollutant description - a description of the CAS or pollutant code contained in the inventory	40-character string

The columns in the INVTABLE are used as follows:

- Inventory data name: Unique name for pollutant or activity that will be stored and processed by SMOKE. Because of I/O API restrictions, these names can only include alphanumeric characters and underscores, and the first character must be a letter. In addition, because SMOKE uses double-underscore as a separator when appending to variable names, no double-underscores are permitted in these names.

- Mode: This three-letter code represents the emissions mode that SMOKE uses to augment the pollutant names. Emissions mode usually refers to the on-road or nonroad mobile exhaust or evaporative emissions. If present in a record of the Inventory Table, SMOKE will combine the Mode field with the Inventory Data Name field using a double underscore. So, if the Inventory Data Name is "VOC" and the Mode is "EXH", then the pollutant name in SMOKE intermediate files will be stored as "EXH__VOC".

- Inventory Pollutant Code: This code represents the pollutant that is provided in the inventory file input into SMOKE, and is sometimes the Chemical Abstracts Service (CAS) number when the inventory uses CAS numbers to identify pollutants. **Smkinven** matches this code to a ORL-formatted inventory file to assign the combination of Mode and Inventory Data Name (columns 2 and 1 as described in the previous bullet), which is used in all subsequent SMOKE intermediate files and reports. This code is not used by SMOKE in any other way.

- SPECIATE4 ID: The SPECIATE4 ID is the unique species identifier in SPECIATE4 (the "ID" field in the SPECIES_PROPERTIES table). This code is not used by SMOKE, but rather is used during import of the Inventory Table to the Speciation Tool.

- Reactivity group: The reactivity group for the ASPEN model. It is not used in SMOKE.

- Keep: Indicates whether SMOKE should keep the CAS number associated with this inventory data name. SMOKE will keep any inventory data with at least one CAS number with a Y.

- Factor: The adjustment factor to use in splitting the emissions associated with a CAS number into different inventory data names, or reducing the mass of a metal compound associated with a CAS number to account for the metal mass only.

- VOC or TOG component: A "V" should be entered if the CAS number is part of VOC emissions or TOG emissions. A "T" should be entered if the CAS number is part of TOG emissions only. An "N" should be entered if the CAS number is part of neither VOC or TOG. Any other value other than "V" or "T" will be treated as an "N".

- Model species: A "Y" should be entered if the pollutant is also a VOC model species for the AQM being targeted for the run. Any other values will be treated as an "N".

- Explicit in mechanism: A "Y" should be entered if (1) the pollutant is a VOC species (the model-species column must be "Y" as well), and (2) that model-species is explicitly treated in the AQM's chemical mechanism. The pollutant can be an explicit species in the chemical mechanism only when the GSPRO file definitions of the lumped VOC species have been adjusted to account for the pollutants/species that are explicit in the mechanism. The current explicit-in-mechanism species being used in the toxics CB4 mechanism for CMAQ are formaldehyde, acetaldheyde, acrolein and 1,3 butadiene.

- Activity: A "Y" should be entered if the inventory data name is a type of activity instead of emissions. Any other values will be treated as an "N".

- NTI: Code identifying a HAP on the Clean Air Act HAP list (list of 188 HAPs). Not currently used by SMOKE; will be used in the future for growing HAP emissions to future years.

- Units: The units to store the data in the SMOKE intermediate files. This should be "short tons/year" for emissions values, "miles/year" for VMT, and "miles/hr" for speed. Although this is provided as an input field to allow possible future changes, these values should *not* be changed by the user and are unrelated to the units of the emissions in the input files.

- Inventory data descriptions: This is the description of the inventory data name listed in columns 1-11.

- CAS pollutant description: Describes the pollutant that is coming from the inventory that is being input into SMOKE.

8.10.5. MACTDESC: MACT code descriptions

This file provides textual descriptions of the MACT codes. The format of the MACTDESC file is shown in Table 8.86, "Format for MACTDESC" [452].

Table 8.86. Format for MACTDESC

Position	Name	Type	Description
A	CODE	Char (6)	MACT code
B	DESC	Char (200)	Description for MACT code

8.10.6. NAICSDESC: NAICS code descriptions

This file provides text descriptions of the NAICS codes. The format of the NAICSDESC file is shown in Table 8.87, "Format for NAICSDESC" [453].

Table 8.87. Format for `NAICSDESC`

Position	Name	Type	Description
A	CODE	Char (6)	NAICS code
B	DESC	Char (200)	Description for NAICS code

8.10.7. `NHAPEXCLUDE`: NonHAP VOC calculation inclusions/exclusions file

The `NHAPEXCLUDE` file provides a way to exclude sources from the NONHAPVOC or NONHAPTOG calculation. For SMOKE versions 2.6 and higher, a new special header in the `NHAPEXCLUDE` file to allow **Smkinven** to either include or exclude sources from NONHAPVOC or NONHAPTOG calculation. When the header is defined as /INCLUDE/, **Smkinven** will treat selected sources as integrated sources; **Smkinven** will use the toxics pollutants from the selected sources to compute NONHAPVOC or NONHAPTOG. The toxic emissions will be retained or dropped by **Smkinven**, depending on the status of the pollutants in the `INVTABLE` file (see Section 8.10.4, "`INVTABLE`: Inventory table" [450]). If the pollutant is a model species and an explicit species, then the toxic emissions will not be dropped. If the pollutant is only a model species but not an explicit species, it will be renamed by appending "_NOI" to the species name. If it is neither a model species nor an explicit species, the toxics mass will be dropped to prevent double counting of the inventory mass. When the header is defined as /EXCLUDE/ which is the default setting, **Smkinven** willl exclude the selected sources from the HAP-CAP integratoin approach. All other inventory sources will be used to compute NONHAPVOC or NONHAPTOG.

The `NHAPEXCLUDE` file is only used when `SMK_PROCESS_HAPS` is set to `PARTIAL`. If `SMK_PRO-CESS_HAPS` is set to `ALL` or `NONE`, the `NHAPEXCLUDE` file is not required (Section 6.13.3.2, "Input Environment Variables" [265]). Once integration/non-integration processing is done, **Smkreport** can create a report that separates integrated and non-integrated sources using the `BY INTEGRATE` instruction (Section 7.3.3, "Create-report Section" [312]).

NOTE: Whenever `SMK_PROCESS_HAPS` is set to process HAPs, a new inventory pollutant, "VOC_INV" or "TOG_INV", will be created to retain the original VOC or TOG values for the purpose of creating quality assurance reports with **Smkreport**.

The `NHAPEXCLUDE` file contains the country/state/county and SCC codes of those sources that SMOKE will include or exclude from the NONHAP calculation. The file format is shown in Table 8.88, "Format for `NHAPEXCLUDE`" [454].

Table 8.88. Format for `NHAPEXCLUDE`

Line	Position	Description
1	A	/INCLUDE/ or /EXCLUDE/
2+	A	Country/state/county code (6-digit integer in format YSSCCC where Y=country, SS=state, and CCC=county • leave blank if not country/state-dependent • set to YSS000 for country/state-dependent • set to YSSCCC for county-dependent
	B	Source Classification Category (SCC) code (20-character maximum)
	C	Facility ID for FF10 Point Sources (Plant ID for ORL), (Character(15), (optional ; leave blank if entry is not plant-specific)
	D	Characteristic 1 (for FF10, this is Unit ID; for ORL, this is Point ID), optional (Character(15))
	E	Characteristic 2 (for FF10, this is Release Point ID, (for ORL, this is stack ID); optional (Character(15))
	F	Characteristic 3 (for FF10, this is Process ID, (for ORL, this is segment ID); optional (Character(15))
	G	Characteristic 4 (for FF10 or ORL, this is 10-digit SCC, or 5-digit with zeroes, or 20-digit maximum SCC) (Character) (optional)
	H	Characteristic 5 (blank for and ORL) (optional)

For area/nonpoint, on-road mobile, and nonroad mobile sources, zeros can be used within the country/state/county code and the SCC code to make an entry more generic. For example, a SCC-only entry would have a zero for the country/state/county code. For a left-2 SCC entry, the final eight characters of the SCC would be zero. The following list illustrates entries for each of the 14 possible cross-reference levels. See Section 2.3.5, "Source Classification Codes" [14] for more information about the different levels within an SCC.

1. 001001 2501080050 (select a specific county and SCC)

2. 001001 2501080000 (select a specific county and all SCCs that match levels 1, 2, and 3)

3. 001001 2501000000 (select a specific county and all SCCs that match levels 1 and 2)

4. 001001 2500000000 (select a specific county and all SCCs that match level 1)

5. 001000 2501080050 (select a whole state and SCC)

6. 001000 2501080000 (select a whole state and all SCCs that match levels 1, 2, and 3)

7. 001000 2501000000 (select a whole state and all SCCs that match levels 1 and 2)

8. 001000 2500000000 (select a whole state and all SCCs that match level 1)

9. 001001 0000000000 (select all SCCs in a specific county)

10. 001000 0000000000 (select all SCCs in a specific state)

11. 000000 2501080050 (select a specific SCC)

12. 000000 2501080000 (select all SCCs that match levels 1, 2, and 3)

13. 000000 2501000000 (select all SCCs that match levels 1 and 2)

14. 000000 2500000000 (select all SCCs that match level 1)

For point sources, the available combinations depend on the source definition. A header is used in the file to indicate the source definition. (The header is required for IDA inventories). For IDA-formatted inventories, a source is defined as the unique combination of country/state/county code, plant ID, point ID, stack ID, segment, and SCC. This documentation applies to IDA-formatted inventories only. "5-digit SCC" means a 10-digit SCC with the last 5 digits set to zero. If only 8-digit SCCs are available, these can be used and SMOKE will assume leading zeros. In this case, the "5-digit SCC" becomes a "3-digit SCC".

The hierarchy of cross-referencing is the following.

1. Country/state/county code, 10-digit SCC, plant ID, point ID, stack ID, segment, and pollutant
2. Country/state/county code, 10-digit SCC, plant ID, point ID, stack ID, and pollutant
3. Country/state/county code, 10-digit SCC, plant ID, point ID, and pollutant
4. Country/state/county code, 10-digit SCC, plant ID, and pollutant
5. Country/state/county code, 10-digit SCC, plant ID, point ID, stack ID, and segment
6. Country/state/county code, 10-digit SCC, plant ID, point ID, and stack ID
7. Country/state/county code, 10-digit SCC, plant ID, and point ID
8. Country/state/county code, 10-digit SCC, and plant ID
9. Country/state/county code, 10-digit SCC, and pollutant
10. Country/state/county code, 5-digit SCC, and pollutant
11. Country/state code, 10-digit SCC, and pollutant
12. Country/state code, 5-digit SCC, and pollutant
13. 10-digit SCC and pollutant
14. 5-digit SCC and pollutant
15. Country/state/county code and 10-digit SCC
16. Country/state/county code and 5-digit SCC
17. Country/state code and 10-digit SCC
18. Country/state code and 5-digit SCC
19. 10-digit SCC
20. 5-digit SCC
21. Country/state/county code
22. Country/state code
23. Ultimate default (all values = 0)

8.10.8. `ORISDESC`: **ORIS description**

The `ORISDESC` file provides descriptive information for the ORIS IDs, which **Smkinven** uses when reporting the status of reading and matching CEM data. The information in this file will appear in the `REPINVEN` output file from **Smkinven**. The format of the `ORISDESC` file is shown in Table 8.89, "Format for `ORISDESC`" [456]. Note that only the first five columns of data need to appear in the file; the rest can be blank because SMOKE does not use any information past column E.

Table 8.89. Format for `ORISDESC`

Line	Position	Description
1+	A	DOE Plant ID (ORIS ID) (6-character string); must match the same field in the IDA-formatted PTINV file.
	B	Country/State/County code (FIPS code) (Char(5))
	C	Latitude (not used by SMOKE), use -9 for missing
	D	Longitude (not used by SMOKE), use -9 for missing
	E	Plant name (80-character string)
	F	Operating company (not used by SMOKE)
	G	EPA Region (not used by SMOKE)
	H	State name (not used by SMOKE)
	I	County name (not used by SMOKE)
	J	North American Electric Reliability Council Region (e.g., ASCC, ECAR, ERCOT, MAAC, MAIN, MAPP, NPCC, SERC, SPP, WSCC) (not used by SMOKE)
	K	Acid rain program phase (not used by SMOKE) • P1 = Phase I Generator, regulation began in 1995-1999 • P1.5 = Phase II Generator, source opted to begin early regulation prior to the year 2000 • P2 = Phase II Generator, generator regulation began in year 2000 • If blank then plant is likely in a nongenerating status (such as retired, deferred, exempt, or future)

8.10.9. `RWC_COUNTY_TEMP`: **County-specific minimum temperature for Residential Wood Combustion (RWC) Equations**

This file allows to county-specific minimum temperature that is required for RWC equations listed at Section 5.3.5, "**Gentpro**" [191] . The format of the `RWC_COUNTY_TEMP` file is shown in Table 8.90, "Format for `RWC_COUNTY_TEMP`" [456].

Table 8.90. Format for `RWC_COUNTY_TEMP`

Position	Name	Type	Description
A	FIPS	Char(5)	FIPS code
B	Value	Real	Minimum temperature

8.10.10. `SCCDESC`: Source category code descriptions

This file provides textual descriptions of the SCCs, which are used by two programs:

- **Smkinven** when reading hourly emissions data (for reporting in the `REPINVEN` output file)

- **Smkreport** when the user requests the SCC description be included in the report.

There are delimited and fixed formats available for the `SCCDESC` file shown in Table 8.91, "Delimited format for `SCCDESC`" [457] and Table 8.92, "Fixed format for `SCCDESC`" [457]. This file should not include any descriptions that have the pipe symbol (|), single quotes, or double quotes. It is okay to include dashes, slashes, semicolons, colons, commas, and periods in descriptions. The format of the `SCCDESC` file is shown in Table 8.91, "Delimited format for `SCCDESC`" [457].

```
#DELIMITED
"10100101","External Combustion Boilers;Electric Generation;Anthracite Coal;Pul
```

Table 8.91. Delimited format for `SCCDESC`

Lines	Position	Description
1+	A	SCC (20-character maximum string)
	B	SCC description (200-character string)

```
#FIXED
  10100101   External Combustion Boilers;Electric Generation;Anthracite Coal;Pul
```

Table 8.92. Fixed format for `SCCDESC`

Lines	Position	Description
1+	1-10	SCC (20-character maximum string)
	101-300	SCC description (200-character string)

8.10.11. `SCCXREF`: Reference SCC input file

This file provides mapping between reference SCC (i.e., SCC8) and full SCC (i.e., SCC10) for **Smkinven** to convert reference SCC-level inventory to full SCC inventory:

Table 8.93. Delimited format for `SCCXREF`

Lines	Position	Description
1+	A	Full SCC (20-character maximum string)
	B	Reference SCC (20-character maximum string)

Example 8.23. SCCXREF example file

```
Full_scc,Reference_scc
"2201110262","2201110200"
"2201110281","2201110200"
"2201110362","2201110300"
"2201110381","2201110300"
"2201110462","2201110400"
"2201110481","2201110400"
"2201110562","2201110500"
"2201110581","2201110500"
"2201210262","2201210200"
"2201210281","2201210200"
....
```

8.10.12. SICDESC: SIC code descriptions

This file provides text descriptions of the SIC codes. There are delimited and fixed formats available for the SICDESC file shown in Table 8.94, "Delimited format for SICDESC" [458] and Table 8.95, "Fixed format for SICDESC" [458].

```
#DELIMITED
"0111","Agricultural Production - Crops, Cash Grains, Wheat"
```

Table 8.94. Delimited format for SICDESC

Position	Name	Type	Description
A	CODE	Char (20)	SIC code
B	DESC	Char (200)	Description for SIC code

```
#FIXED
0111 Agricultural Production - Crops, Cash Grains, Wheat
```

Table 8.95. Fixed format for SICDESC

Position	Name	Type	Description
1-4	CODE	Char (20)	SIC code
5-205	DESC	Char (200)	Description for SIC code

Example 8.24.

8.10.13. SOURCE_GROUPS: Source apportionment groups

The SOURCE_GROUPS file is used to define groups of sources when processing emissions for source apportionment. See Section 4.4.23, "Output source apportionment data" [175] for general information about using source apportionment in SMOKE. **Movesmrg** and **Smkmerge** read this file when SMK_SRCGROUP_OUTPUT_YN is set to Y.

When reading the SOURCE_GROUPS file, SMOKE will skip lines with data that doesn't apply to the current source category. Lines that contain a plant ID or point ID will be skipped when processing area, biogenic, or mobile sources. Lines containing an SCC will be skipped when processing biogenic sources.

If SMOKE encounters lines with duplicate grouping information (i.e. FIPS, SCC, point source characteristics), an error will be printed to the log and the program will quit.

Multiple grouping criteria can be assigned to the same source group ID by using additional lines. In the example below, all sources with SCCs 2102001000, 2102002000, and 2102004000 will be assigned to source group 1.

```
1 0 2102001000
1 0 2102002000
1 0 2102004000
```

If the source grouping file contains a line with all blank or zero source characteristics, SMOKE will use the given source group number as the default group rather than zero. For example, if the following line is included in the SOURCE_GROUPS file, the default source group number will be 40 instead of 0.

```
40 0 0 0 0  # default group number will be 40 instead of 0
```

The format of the SOURCE_GROUPS file is shown in Table 8.96, "Format for SOURCE_GROUPS" [459].

Table 8.96. Format for SOURCE_GROUPS

Line	Position	Description
1+	A	Source group number, zero is reserved for default source group (Integer) (required)
	B	Country/state/county code (6-digit Character YSSCCC) (optional): • leave blank or enter zero if not country/state-dependent • set to YSS000 for country/state-dependent • set to YSSCCC for county-dependent
	C	20-digit maximum SCC (Character) (optional, set to zero if not SCC-dependent)
	D	Plant ID (Character) (optional; leave blank for not plant-specific)
	E	Point ID (Character) (optional; leave blank for not point-specific)

8.10.14. VNAMMAP: Variable name mapping file

This file is used to rename variables output from **Smk2emis**, which converts the I/O API format to the 2-D-UAM "EMISSIONS" file format. The VNAMMAP file lists the SMOKE variable names and the variable names to be output from **Smk2emis**. It is needed for SMOKE support of UAM-AERO, which requires variable names that cannot be supported by the I/O API format. More information on **Smk2emis** is provided in Section 5.3.12, "**Smk2emis**" [205].

The format of the VNAMMAP file is shown in Table 8.97, "Format for VNAMMAP" [460].

Table 8.97. Format for `VNAMMAP`

Lines	Position	Description
1+	A	SMOKE I/O API variable name
	B	**Smk2emis** output name to be written to the UAM "EMISSIONS" file binary format

Chapter 9. SMOKE Intermediate Files

Table of Contents

9.1. Introduction

An intermediate file is a file that is output from one SMOKE program and input to another SMOKE program. All other files output from programs are considered output files; these are documented in Chapter 10, *SMOKE Output Files* [483].

In this chapter, the intermediate files are organized under headings named for the SMOKE programs that create the files; the files are listed alphabetically within each program section. The order of the programs within the chapter is also alphabetical. Please note that some file types have multiple file names, one per source category. For example, the multiplicative control matrix file name is ACMAT, MCMAT, or PCMAT, depending on whether the area, mobile, or point source category is being processed by **Cntlmat**. In these cases of multiple names for a single file type, the file description is listed one time for all intermediate files.

9.2. Cntlmat

The **Cntlmat** program is responsible for creating the growth and control matrices. **Cntlmat** creates the following intermediate files:

- Multiplicative control matrix: Section 9.2.1, "ACMAT, MCMAT, and PCMAT" [461]

- Growth matrix: Section 9.2.2, "APMAT, MPMAT, and PPMAT" [462]

- Reactivity control matrix: Section 9.2.3, "ARMAT, MRMAT, and PRMAT" [462]

9.2.1. ACMAT, MCMAT, and PCMAT

Type: I/O API NetCDF - adapted gridded format

Used by: **Grwinven**, **Smkmerge**, and **Smkreport**

Table 9.1. I/O API Header for `ACMAT`, `MCMAT`, and `PCMAT`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources in the inventory	NROWS3D from the `AREA`, `MOBL`, or `PNTS` file
NVARS3D	Number of pollutants that have control factors	The contents of the `GCNTL` file indicate which pollutants will be controlled.

Table 9.2. Variables in `ACMAT`, `MCMAT`, and `PCMAT`

Variable Name	Type	Units	Description
Name of pollutant 1	Real	Unitless	Adjustment factor for pollutant 1
...	...	...	...
Name of pollutant NVARS3D	Real	Unitless	Adjustment factor for pollutant NVARS3D

9.2.2. `APMAT`, `MPMAT`, and `PPMAT`

Type: I/O API NetCDF - adapted gridded format

Used by: **Grwinven**, **Smkmerge**, **Smkreport**

Table 9.3. I/O API Header for `APMAT`, `MPMAT`, and `PPMAT`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources in the inventory	NROWS3D from the `AREA`, `MOBL`, or `PNTS` file

Table 9.4. Variables in `APMAT`, `MPMAT`, and `PPMAT`

Variable Name	Type	Units	Description
pfac	Real	Unitless	Projection factor

9.2.3. `ARMAT`, `MRMAT`, and `PRMAT`

Type: I/O API NetCDF - adapted gridded format

Used by: **Grwinven** and **Smkmerge**

Table 9.5. I/O API Header for `ARMAT`, `MRMAT`, and `PRMAT`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of reactivity sources	NSREAC from the GENREACT subroutine
NVARS3D	Number of reactivity species plus the four base variables	The contents of the GSPRO file
NTHIK3D	The number of SMOKE sources	NSRC from the **Smkinven** program

Table 9.6. Variables in `ARMAT`, `MRMAT`, and `PRMAT`

Variable Name	Type	Units	Description
SRCID	Integer	Unitless	Inventory source ID (position)
REPEMIS	Real	UNITS from the AREA, MOBL, or PNTS file	Reactivity base-year emissions
PRJFAC	Real	Unitless	Reactivity projection factor
MKTPEN	Real	Fraction	Reactivity control market penetration
Name of species 1	Real	g/ton or mol/ton	Inventory pollutant combined with model species name
...	...	...	...
Name of species NVARS3D	Real	g/ton or mol/ton	Inventory pollutant combined with model species name

9.3. Elevpoint

The **Elevpoint** program is responsible for selecting the elevated sources. **Elevpoint** creates the following intermediate files:

- Elevated-point-source identification file: Section 9.3.1, "`PELV`" [463]

- Stack groups file: Section 9.3.2, "`STACK_GROUPS`" [464]

9.3.1. `PELV`

Type: ASCII - list-directed

Used by: **Laypoint**, **Smkmerge**, and **Smkreport**

Table 9.7. File columns in `PELV`

Line	Column	Type	Units	Description
1+	A	Integer	Unitless	Source number for elevated source, zero if a PinG source
	B	Integer	Unitless	Source number for PinG source, zero if simply an elevated source	
	C	Integer	Unitless	Group number for source	

9.3.2. `STACK_GROUPS`

Type: I/O API NetCDF - adapted gridded format

Used by: **Laypoint** and **Smkmerge**

Table 9.8. I/O API Header for `STACK_GROUPS`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of PinG groups	Determined by the `PELVCONFIG` file

Table 9.9. Variables in `STACK_GROUPS`

Variable Name	Type	Units	Description
ISTACK	Integer	Unitless	Stack group number
LATITUDE	Real	degrees	Latitude
LONGITUDE	Real	degrees	Longitude
STKDM	Real	m	Inside stack diameter
STKHT	Real	m	Stack height above ground surface
STKTK	Real	K	Stack exit temperature
STKVE	Real	m/s	Stack exit velocity
STKFLW	Real	m^3/s	Stack exit flow rate
STKCNT	Integer	Unitless	Number of stacks in group
ROW	Integer	Unitless	Grid row number
COL	Integer	Unitless	Grid column number
XLOCA	Real	Coordinate system projection units	Projection x coordinate
YLOCA	Real	Coordinate system projection units	Projection y coordinate

9.4. Grdmat

The **Grdmat** program is responsible for creating the gridding matrix. **Grdmat** creates the following intermediate files:

• Gridding matrix file: Section 9.4.1, "`AGMAT`, `MGMAT`, and `PGMAT`" [465]

9.4.1. `AGMAT`, `MGMAT`, and `PGMAT`

Type: I/O API NetCDF - adapted gridded format

Used by: **Smkmerge** and **Smkreport**

Table 9.10. I/O API Header for `AGMAT`, `MGMAT`, and `PGMAT`

I/O API Header variable	Meaning	Set by...
NCOLS3D	Number of source-cell intersections	Determined by the grid and the sources in the inventory
NROWS3D	Number of cells in the grid	Number of columns times number of rows in the grid

Table 9.11. Variables in `AGMAT`, `MGMAT`, and `PGMAT`

Variable Name	Type	Units	Description
COEFF-INDEX	Real	Unitless	Source gridding coefficients

9.5. Gentpro

The **Gentpro** program is responsible for creating the monthly, daily or hourly temporal profiles for **Temporal**. **Gentpro** creates the following intermediate files depending on the options:

• Temporal cross-reference file: Section 8.3.2, "`[A|M|P]TREF`: Temporal cross-reference file" [383]

• Monthly temporal profile file: Section 9.5.1, "`TPRO_MON`" [465]

• Daily temporal profile file: Section 9.5.2, "`TPRO_DAY`" [466]

• Hourly temporal profile file: Section 9.5.3, "`TPRO_HOUR`" [467]

9.5.1. `TPRO_MON`

Type: ASCII - list-directed

Used by: **Temporal**

Table 9.12. File columns in `TPRO_MON`

Line	Column	Type	Units	Description
1+	A	Integer	Unitless	Temporal profile ID #1 (=FIPS #1)
	B	Real	Unitless	Fraction of annual emission in January
	C	Real	Unitless	Fraction of annual emission in February
	...	Real	Unitless	...
	M	Real	Unitless	Fraction of annual emission in December
2+	A	Integer	Unitless	Temporal profile ID #2 (=FIPS #2)
	B	Real	Unitless	Fraction of annual emission in January
	C	Real	Unitless	Fraction of annual emission in February
	...	Real	Unitless	...
	M	Real	Unitless	Fraction of annual emission in December
...	...	...	...	...

9.5.2. `TPRO_DAY`

Type: ASCII - list-directed

Used by: **Temporal**

Table 9.13. File columns in `TPRO_DAY`

Line	Column	Type	Units	Description
1+	1	Integer	Unitless	Temporal profile ID #1 (=FIPS #1)
	2	Integer	Unitless	Month (i.e., January = 1)
	3	Real	Unitless	Fraction of January monthly total emission on Day 1
	...	Real	Unitless	...
	33	Real	Unitless	Fraction of January monthly total emission on Day 31
2+	1	Integer	Unitless	Temporal profile ID #2 (=FIPS #2)
	2	Real	Unitless	Month (i.e., February = 2)
	3	Real	Unitless	Fraction of February monthly total emission on Day 1
	...	Real	Unitless	...
	30	Real	Unitless	Fraction of February monthly total emission on Day 28
...	...	...	...	...
...	...	...	...	...
100+	1	Integer	Unitless	Temporal profile ID #100 (=FIPS #100)
	2	Integer	Unitless	Month (i.e., January = 1)
	3	Real	Unitless	Fraction of January monthly total emission on Day 1
	...	Real	Unitless	...
	33	Real	Unitless	Fraction of January monthly total emission on Day 31
...	...	...	...	...

9.5.3. `TPRO_HOUR`

Type: I/O API NetCDF - adapted gridded format

Used by: **Temporal**

Table 9.14. I/O API Header for `TPRO_HOUR`

I/O API Header variable	Meaning
NROWS3D	Number of Sources (=Counties)
NVARS3D	Number of temporal profiles

Table 9.15. Variables in `TPRO_HOUR`

Variable Name	Type	Units	Description
FIPS	Integer	Unitless	Temporal profile ID (=FIPS code)
HRLSRC	Real	Unitless	Hourly total by source
DAYTOT	Real	Unitless	Daily total by source
MONTOT	Real	Unitless	Monthly total by source
ANNTOT	Real	Unitless	Annual total by source

9.6. Grwinven

The **Grwinven** program is responsible for applying growth and controls to the inventory. **Grwinven** creates the following intermediate files:

- Grown inventory file: Section 9.6.1, "`AREA_O`, `MOBL_O` and `PNTS_O`" [468]

- Grown IDA/ORL inventory file: Section 9.6.2, "`ARINV_O`, `MBINV_O`, `MBINV_AO`, `PTINV_O`" [470]

9.6.1. `AREA_O`, `MOBL_O` and `PNTS_O`

Type: I/O API NetCDF - adapted gridded format

Used by: **Cntlmat**, **Elevpoint**, **Grdmat**, **Grwinven**, **Laypoint**, **Smkmerge**, **Spcmat**, and **Temporal**

Table 9.16. I/O API Header for `AREA_O`, `MOBL_O`, and `PNTS_O`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources	NROWS3D from the `AREA`, `MOBL`, or `PNTS` file
NVARS3D	Number of variables in the file	Based on source category

Table 9.17. Variables in `AREA_O`, `MOBL_O`, and `PNTS_O`

Variable Name	Type	Units	Description
IFIP	Integer	Unitless	State and county FIPS code
TZONES	Integer	Unitless	Time zone for site
TPFLAG	Integer	Unitless	Use week, month temporal profiles or not
INVYR	Integer	Unitless	Year of inventory for this record
CELLID	Integer	Unitless	Cell number
IRCLAS	Integer	Unitless	Roadway type (mobile only)
IVTYPE	Integer	Unitless	Vehicle type code (mobile only)
XLOC1	Integer	Unitless	Longitude at beginning of link (mobile only)
YLOC1	Integer	Unitless	Latitude at beginning of link (mobile only)
XLOC2	Integer	Unitless	Longitude at end of link (mobile only)
YLOC2	Integer	Unitless	Latitude at end of link (mobile only)
ISIC	Integer	Unitless	SIC code (area or point only)
XLOCA	Real	degrees	Longitude (area or point only)
YLOCA	Real	degrees	Latitude (area or point only)
STKHT	Real	m	Stack height (point only)
STKDM	Real	ms	Stack diameter (point only)
STKTK	Real	K	Stack exhaust temperature (point only)
STKVE	Real	m/s	Stack exit velocity (point only)
Name of pollutant 1>	Real	ton/yr	Annual emissions for pollutant 1
OS_<Pollutant name 1>	Real	ton/day	Ozone season emissions for pollutant 1
EF_<Pollutant name 1>	Real	SCC units	Emission factor for pollutant 1
CE_<Pollutant name 1>	Real	Unitless	Control efficiency fraction for pollutant 1
RE_<Pollutant name 1>	Real	Unitless	Rule effectiveness fraction for pollutant 1
RP_<Pollutant name 1>	Real	Unitless	Rule penetration fraction for pollutant 1
...	...	...	...
Name of pollutant NVARS3D	Real	ton/yr	Annual emissions for pollutant NVARS3D
OS_<Pollutant name NVARS3D>	Real	ton/day	Ozone season emissions for pollutant NVARS3D
EF_<Pollutant name NVARS3D>	Real	SCC units	Emission factor for pollutant NVARS3D
CE_<Pollutant name NVARS3D>	Real	Unitless	Control efficiency fraction for pollutant NVARS3D
RE_<Pollutant name NVARS3D>	Real	Unitless	Rule effectiveness fraction for pollutant NVARS3D
RP_<Pollutant name NVARS3D>	Real	Unitless	Rule penetration fraction for pollutant NVARS3D

9.6.2. `ARINV_O`, `MBINV_O`, `MBINV_AO`, `PTINV_O`

Type: ASCII - column-specific

Used by: **Smkinven**

See Chapter 8, *SMOKE Input Files* [345] for the formats of the `ARINV`, `MBINV`, and `PTINV` files.

9.7. Laypoint

The **Laypoint** program is responsible for computing layer fractions for point sources. **Laypoint** creates the following intermediate files:

- Layer fractions file: Section 9.7.1, "`PLAY` and `PLAY_EX`" [470]

9.7.1. `PLAY` and `PLAY_EX`

Type: I/O API NetCDF - adapted gridded format

Used by: **Smkmerge** and **Smkreport**

Table 9.18. I/O API Header for `PLAY` and `PLAY_EX`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of point sources or number of explicit sources	Number of sources in the `PNTS` file, or number of explicit sources in the `PHOUR` file

Table 9.19. Variables in `PLAY` and `PLAY_EX`

Variable Name	Type	Units	Description
INDXH	Integer	Unitless	Source number (for explicit sources only)
LFRAC	Real	Unitless	Fraction of plume emitted into layer

9.8. Met4moves

The **Met4moves** program is responsible for creating inputs for the MOVES Driver Script and the SMOKE **Movesmrg** program. **Met4moves** creates the following intermediate files:

- Section 9.8.1, "`MOVES_OUTFILE`" [470]

- Section 9.8.2, "`MOVES_RH_OUTFILE`" [471]

- Section 9.8.3, "`SMOKE_OUTFILE(=METMOVES)`" [471]

9.8.1. `MOVES_OUTFILE`

Type: ASCII - column-specific

Used by: MOVES Driver Script

The output file created by **Met4moves** for RPP mode in the MOVES driver script contains the absolute minimum and maximum temperatures, and sets of 24 hour diurnal temperature profiles based on combinations of min/max temperature bins for all reference counties by temperature bin.

Table 9.20. File columns in `MOVES_OUTFILE`

Column	Type	Units	Description
A	Integer	Unitless	Reference County
B	Integer	Unitless	Fuel Month
C	Character	Unitless	temperatureProfileID
D	Real	Fahrenheit	Temperature at hour 0
E	Real	Fahrenheit	Temperature at hour 1
F	Real	Fahrenheit	Temperature at hour 2
G	Real	Fahrenheit	Temperature at hour 3
...	...	...	...
AC	Real	Fahrenheit	Temperature at hour 23

9.8.2. `MOVES_RH_OUTFILE`

Type: ASCII - column-specific

Used by: MOVES Driver Script

The output file created by **Met4moves** program for RPD, RPV and RPH modes in both MOVES and SMOKE models contains fuel month-specific min and max temperatures, and averaged RH values in local time for all reference counties by temperature bin.

Table 9.21. File columns in `MOVES_RH_OUTFILE`

Column	Type	Units	Description
A	Integer	Unitless	Reference County
B	Integer	Unitless	Fuel Month
C	Real	%	Average Relative Humidity
D	Real	Fahrenheit	Lowest Minimum Temperature across all inventory counties
E	Real	Fahrenheit	Highest Maximum Temperature across all inventory counties
F	Real	Fahrenheit	Temperature bin

9.8.3. `SMOKE_OUTFILE(=METMOVES)`

Type: I/O API NetCDF - adapted gridded format

Used by: **Movesmrg** (if RPP_MODE=Y).

The output file created by **Met4moves** program for use by **Movesmrg**. It contains gridded daily min/max temperatures for every inventory county. **Met4moves** adjusts the native MCIP time zone (GMT) to local time in order to properly use the SMOKE-ready lookup tables, which are given in local time.

Type: I/O API NetCDF - adapted gridded format

Used by: **Smkmerge** and **Smkreport**

Table 9.22. I/O API Header for `SMOKE_OUTFILE`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources	Determined by `MOBL` file
NVARS3D	Number of temperature variables	Determined by `MOBL` file

Table 9.23. Variables in `SMOKE_OUTFILE`

Variable Name	Type	Units	Description
MINTEMP	Real	Fahrenheit	Daily minimum temperature
MAXTEMP	Real	Fahrenheit	Daily maximum temperature

9.9. Movesmrg

The **Movesmrg** program is responsible for creating model-ready emissions files. **Smkmerge** creates the following intermediate file:

- Model-ready emissions file: Section 9.9.1, "`[M][G][T][S][_S][_L]`" [472]

9.9.1. `[M][G][T][S][_S][_L]`

Type: I/O API NetCDF - adapted gridded format

Used by: **Mrggrid**

9.10. Normbeis3

The **Normbeis3** program is responsible for computing normalized biogenic emissions. **Normbeis3** creates the following intermediate file:

- Normalized biogenic emissions file: Section 9.10.1, "`B3GRD`" [472]

9.10.1. `B3GRD`

Type: I/O API NetCDF - adapted gridded format

Used by: **Tmpbeis3**

Table 9.24. I/O API Header for `B3GRD`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of rows in the grid	NROWS3D from the gridded land use file
NCOLS3D	Number of columns in the grid	NCOLS3D from the gridded land use file
NVARS3D	Number of biogenic emission species times the number of seasons modeled	`B3DIMS3.EXT` include file

Table 9.25, "Variables in `B3GRD` for BEIS version 3.14" [474] contains names and descriptions of variables in the `B3GRD` output file form **Normbeis3**, BEIS3.

For BEIS version 3.14, **Normbeis3** calculates normalized emissions for both the summer and winter seasons. For nitric acid (NO) calculations, however, emissions depend on a given grid cell's land use category (or categories). For agricultural land use portions of a grid cell, a determination is made of whether the given day is in the growing season or the non-growing season. The appropriate emissions factor is then applied to the NO emissions calculations. The agricultural land use categories USGS Cropgrass (a cropland/grassland mosaic) and USGS Cropwoodland (a cropland/woodland mosaic) represent grid cells with mixed vegetation and only half the area is assumed to be covered by crops. Accordingly, the corresponding emissions factors for these two categories are multiplied by 0.5 for NO calculations. (Note that the USGS Drycrop and USGS Irrcrop land use categories assume the vegetation is all crop.) For the non-growing season in agricultural areas, the emissions factor for the USGS Grassland land use category is used for NO emission calculations. Finally, for non-agricultural areas, the USGS Grassland emissions factor is used all year round. Table 9.21b contains the names and descriptions of variables in the B3GRD output file from **Normbeis3**, BEIS version 3.14

Table 9.25. Variables in B3GRD **for BEIS version 3.14**

Variable Name	Type	Units	Description
AVG_ISOP<Season (S or W)>	Real	g C/km^2-hr	Normalized isoprene emissions
AVG_MBO<Season (S or W)>	Real	g C/km^2-hr	Normalized 2-methyl-3-buten-2-ol emissions
AVG_METH<Season (S or W)>	Real	g C/km^2-hr	Normalized methanol emissions
AVG_APIN<Season (S or W)>	Real	g C/km^2-hr	Normalized alpha-pinene emissions
AVG_BPIN<Season (S or W)>	Real	g C/km^2-hr	Normalized beta-pinene emissions
AVG_D3CAR<Season (S or W)>	Real	g C/km^2-hr	Normalized delta-3-carene emissions
AVG_DLIM<Season (S or W)>	Real	g C/km^2-hr	Normalized d-limonene emissions
AVG_CAMPH<Season (S or W)>	Real	g C/km^2-hr	Normalized camphene emissions
AVG_MYRC<Season (S or W)>	Real	g C/km^2-hr	Normalized myrcene emissions
AVG_ATERP<Season (S or W)>	Real	g C/km^2-hr	Normalized alpha-terpinene emissions
AVG_BPHE<Season (S or W)>	Real	g C/km^2-hr	Normalized beta-phellandrene emissions
AVG_SABI<Season (S or W)>	Real	g C/km^2-hr	Normalized sabinene emissions
AVG_PCYM<Season (S or W)>	Real	g C/km^2-hr	Normalized p-cymene emissions
AVG_OCIM<Season (S or W)>	Real	g C/km^2-hr	Normalized ocimene emissions
AVG_ATHU<Season (S or W)>	Real	g C/km^2-hr	Normalized alpha-thujene emissions
AVG_TRPO<Season (S or W)>	Real	g C/km^2-hr	Normalized terpinolene emissions
AVG_GTERP<Season (S or W)>	Real	g C/km^2-hr	Normalized gamma-terpinene emissions
AVG_ETHE<Season (S or W)>	Real	g C/km^2-hr	Normalized ethene emissions
AVG_PROPE<Season (S or W)>	Real	g C/km^2-hr	Normalized propene emissions
AVG_ETHO<Season (S or W)>	Real	g C/km^2-hr	Normalized ethanol emissions
AVG_ACET<Season (S or W)>	Real	g C/km^2-hr	Normalized acetone emissionS
AVG_HEXA<Season (S or W)>	Real	g C/km^2-hr	Normalized hexanal emissions
AVG_HEXE<Season (S or W)>	Real	g C/km^2-hr	Normalized hexenol emissions

Variable Name	Type	Units	Description
AVG_HEXY<Season (S or W)>	Real	g C/km^2-hr	Normalized hexenylacetate emissions
AVG_FORM<Season (S or W)>	Real	g C/km^2-hr	Normalized formaldehyde emissions
AVG_ACTAL<Season (S or W)>	Real	g C/km^2-hr	Normalized acetaldehyde emissions
AVG_BUTE<Season (S or W)>	Real	g C/km^2-hr	Normalized butene emissions
AVG_ETHA<Season (S or W)>	Real	g C/km^2-hr	Normalized ethane emissions
AVG_FORAC<Season (S or W)>	Real	g C/km^2-hr	Normalized formic acid emissions
AVG_ACTAC<Season (S or W)>	Real	g C/km^2-hr	Normalized acetic acid emissions
AVG_BUTO<Season (S or W)>	Real	g C/km^2-hr	Normalized butenone (aka methyl vinyl ketone) emissions
AVG_CO<Season (S or W)>	Real	g C/km^2-hr	Normalized carbon monoxide emissions
AVG_ORVOC<Season (S or W)>	Real	g C/km^2-hr	Normalized other reactive VOC emissions
LAI_ISOP<Season (S or W)>	Real	m^2/m^2	Leaf area index for isoprene emissions
LAI_MBO<Season (S or W)>	Real	m^2/m^2	Leaf area index for 2-methyl-3-buten-2-ol emissions
LAI_METH<Season (S or W)>	Real	m^2/m^2	Leaf area index for methanol emissions
AVG_NOAG_GROW	Real	g N/km^2-hr	Normalized nitric acid emissions in agricultural areas during the growing season
AVG_NOAG_NONGROW	Real	g N/km^2-hr	Normalized nitric acid emissions in agricultural areas during the non-growing season
AVG_NONONAG	Real	g N/km^2-hr	Normalized nitric acid emissions in non-agricultural areas

9.11. Smkinven

The **Smkinven** program is responsible for importing the raw inventory into SMOKE format. **Smkinven** creates the following intermediate files:

- Average inventory file: Section 9.11.1, "`AREA`, `MOBL`, and `PNTS`" [475]

- Source description file: Section 9.11.2, "`ASRC`, `MSRC`, and `PSRC`" [478]

- Period-specific inventory file: Section 9.11.3, "`PDAY`, `PHOUR`, `ARDAY`, and `ARHOUR`" [478]

9.11.1. `AREA`, `MOBL`, and `PNTS`

Type: I/O API NetCDF - adapted gridded format

Used by: **Cntlmat**, **Elevpoint**, **Grdmat**, **Grwinven**, **Laypoint**, **Smkmerge**, **Smkreport**, **Spcmat**, and **Temporal**

Table 9.26. I/O API Header for `AREA`, `MOBL`, and `PNTS`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources	NROWS3D from the `AREA`, `MOBL`, or `PNTS` file
NVARS3D	Number of variables in the file	Based on source category

Table 9.27. Variables in `AREA`, `MOBL`, and `PNTS`

Variable Name	Type	Units	Description
IFIP	Integer	Unitless	State and county FIPS code
TZONES	Integer	Unitless	Time zone for site
TPFLAG	Integer	Unitless	Use week, month temporal profiles or not
INVYR	Integer	Unitless	Year of inventory for this record
CELLID	Integer	Unitless	Cell number
IRCLAS	Integer	Unitless	Roadway type (mobile only)
IVTYPE	Integer	Unitless	Vehicle type code (mobile only)
XLOC1	Integer	Unitless	Longitude at beginning of link (mobile only)
YLOC1	Integer	Unitless	Latitude at beginning of link (mobile only)
XLOC2	Integer	Unitless	Longitude at end of link (mobile only)
YLOC2	Integer	Unitless	Latitude at end of link (mobile only)
ISIC	Integer	Unitless	SIC code (area or point only)
XLOCA	Real	degrees	Longitude (area or point only)
YLOCA	Real	degrees	Latitude (area or point only)
STKHT	Real	m	Stack height (point only)
STKDM	Real	ms	Stack diameter (point only)
STKTK	Real	K	Stack exhaust temperature (point only)
STKVE	Real	m/s	Stack exit velocity (point only)
Name of pollutant 1>	Real	ton/yr	Annual emissions for pollutant 1
OS_<Pollutant name 1>	Real	ton/day	Ozone season emissions for pollutant 1
EF_<Pollutant name 1>	Real	SCC units	Emission factor for pollutant 1
CE_<Pollutant name 1>	Real	Unitless	Control efficiency fraction for pollutant 1
RE_<Pollutant name 1>	Real	Unitless	Rule effectiveness fraction for pollutant 1
RP_<Pollutant name 1>	Real	Unitless	Rule penetration fraction for pollutant 1
...	...	...	...
Name of pollutant NVARS3D	Real	ton/yr	Annual emissions for pollutant NVARS3D
OS_<Pollutant name NVARS3D>	Real	ton/day	Ozone season emissions for pollutant NVARS3D
EF_<Pollutant name NVARS3D>	Real	SCC units	Emission factor for pollutant NVARS3D
CE_<Pollutant name NVARS3D>	Real	Unitless	Control efficiency fraction for pollutant NVARS3D
RE_<Pollutant name NVARS3D>	Real	Unitless	Rule effectiveness fraction for pollutant NVARS3D
RP_<Pollutant name NVARS3D>	Real	Unitless	Rule penetration fraction for pollutant NVARS3D

9.11.2. `ASRC`, `MSRC`, **and** `PSRC`

Type: ASCII - list-directed

Used by: **Cntlmat**, **Elevpoint**, **Grdmat**, **Grwinven**, **Laypoint**, **Smkmerge**, **Smkreport**, **Spcmat**, and **Temporal**

Table 9.28. File columns in `ASRC`, `MSRC`, and `PSRC`

Line	Column	Type	Units	Description
1	A	Integer	Unitless	Number of columns of data (N)
	B	Character	Unitless	Format statement for ASCII data	
2	A	Character	Unitless	Description of column 1
...	...	...	...	...
N+1	A	Character	Unitless	Description of column N
(N+2)+	A	Integer	Unitless	See column 1 header description
...	...	...	...	
	Character	Unitless	See column N header description	

9.11.3. `PDAY, PHOUR, ARDAY,` **and** `ARHOUR`

Type: I/O API NetCDF - adapted gridded format

Used by: **Laypoint** and **Temporal**

Table 9.29. I/O API Header for `PDAY, PHOUR, ARDAY,` and `ARHOUR`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of period-specific sources	`PTDAY` or `PTHOUR` or `ARDAY` or `ARHOUR` files read in by **Smkinven**
NVARS3D	Number of pollutants + 1	`PTDAY` or `PTHOUR` or `ARDAY` or `ARHOUR` files read in by **Smkinven**

Table 9.30. Variables in `PDAY, PHOUR, ARDAY,` and `ARHOUR`

Variable Name	Type	Units	Description
INDX<D or H>	Integer	Unitless	Source number
Name of pollutant 1	Real	ton/h	Period-specific emissions for pollutant 1
...	...	...	...
Name of pollutant NVARS3D	Real	ton/h	Period-specific emissions for pollutant NVARS3D

9.12. Smkmerge

The **Smkmerge** program is responsible for creating model-ready emissions files. **Smkmerge** creates the following intermediate file:

- Model-ready emissions file: Section 9.12.1, "`[A|B|M|P|E][G][T][S][3D][_S][_L]`" [479]

9.12.1. `[A|B|M|P|E][G][T][S][3D][_S][_L]`

Type: I/O API NetCDF - adapted gridded format

Used by: **Mrggrid** and **Smkmerge**

These files can be model-ready emissions files by source category or all categories. See Section 10.6.1, "`[A|B|M|P|E]G[T][S][3D][_S][_L]`" [489] for format information.

9.13. Spcmat

The **Spcmat** program is responsible for creating speciation matrices. **Spcmat** creates the following intermediate files:

- Speciation matrix file (mass-based): Section 9.13.1, "`ASMAT_S`, `MSMAT_S`, and `PSMAT_S`" [479]

- Speciation matrix file (mole-based): Section 9.13.2, "`ASMAT_L`, `MSMAT_L`, and `PSMAT_L`" [479]

9.13.1. `ASMAT_S`, `MSMAT_S`, and `PSMAT_S`

Type: I/O API NetCDF - adapted gridded format

Used by: **Smkmerge** and **Smkreport**

Table 9.31. I/O API Header for `ASMAT_S`, `MSMAT_S`, and `PSMAT_S`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources	Determined by AREA, MOBL, or PNTS file
NVARS3D	Number of inventory pollutants combine with output species	Determined by the GSPRO file

Table 9.32. Variables in `ASMAT_S`, `MSMAT_S`, and `PSMAT_S`

Variable Name	Type	Units	Description
SVAR001	Real	g/ton	Mass-speciation coefficient
...	...	...	...
SVAR<NVARS3D>	Real	g/ton	Mass-speciation coefficient

9.13.2. `ASMAT_L`, `MSMAT_L`, and `PSMAT_L`

Type: I/O API NetCDF - adapted gridded format

Used by: **Smkmerge** and **Smkreport**

Table 9.33. I/O API Header for `ASMAT_L`, `MSMAT_L`, and `PSMAT_L`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources	Determined by `AREA`, `MOBL`, or `PNTS` file
NVARS3D	Number of inventory pollutants combine with output species	Determined by the `GSPRO` file

Table 9.34. Variables in `ASMAT_L`, `MSMAT_L`, and `PSMAT_L`

Variable Name	Type	Units	Description
SVAR001	Real	mol/ton	Mole-speciation coefficient
...	...	...	...
SVAR<NVARS3D>	Real	mol/ton	Mole-speciation coefficient

9.14. Temporal

The **Temporal** program is responsible for generating hourly emissions by source. **Temporal** creates the following intermediate file:

- Hourly emissions file: Section 9.14.1, "`ATMP`, `MTMP`, and `PTMP`" [480]

9.14.1. `ATMP`, `MTMP`, and `PTMP`

Type: I/O API NetCDF - adapted gridded format

Used by: **Smkmerge** and **Smkreport**

Table 9.35. I/O API Header for `ATMP`, `MTMP`, and `PTMP`

I/O API Header variable	Meaning	Set by...
NROWS3D	Number of sources	Determined by `AREA`, `MOBL`, or `PNTS` file
NVARS3D	Number of inventory pollutants	Determined by `AREA`, `MOBL`, or `PNTS` file

Table 9.36. Variables in `ATMP`, `MTMP`, and `PTMP`

Variable Name	Type	Units	Description
Name of pollutant 1	Real	ton/h	Hourly emissions for pollutant 1
...	...	...	...
Name of pollutant NVARS3D	Real	ton/h	Hourly emissions for pollutant NVARS3D

9.15. Tmpbeis3

The **Tmpbeis3** is responsible for creating gridded, hourly biogenic emissions. **Tmpbeis3** creates the following intermediate files:

- Model-ready biogenic emissions: Section 9.15.1, "`B3GTS_L` and `B3GTS_S`" [481]

9.15.1. `B3GTS_L` and `B3GTS_S`

Type: I/O API NetCDF - adapted gridded format

Used by: **Mrggrid** and **Smkmerge**

These files are model-ready output, but can be input into **Mrggrid** or **Smkmerge** to be combined with other source category model-ready files. See Section 10.7.1, "`B3GTS_L` and `B3GTS_S`" [489] for format information.

Chapter 10. SMOKE Output Files

Table of Contents

10.1. Introduction

An output file is a file that is output from a SMOKE program and not used as input to any other SMOKE programs. This include the model-ready emissions files and various reports. Files that are used as input to other SMOKE programs are considered intermediate files and are documented in Chapter 9, *SMOKE Intermediate Files* [461].

In this chapter, the output files are organized under headings named for the SMOKE programs that create the files; the files are listed alphabetically within each program section. The order of the programs within the chapter is also alphabetical. Please note that some file types have multiple file names, one per source category. For example, the multiplicative control report file name is `ACREP`, `MCREP`, or `PCREP`, depending on whether the area, mobile, or point source category is being processed by **Cntlmat**. In these cases of multiple names for a single file type, the file description is listed one time for all output files.

10.2. Cntlmat

The **Cntlmat** program is responsible for creating the growth and control matrices. **Cntlmat** creates the following output files:

- Multiplicative control report: Section 10.2.1, "`ACREP`, `MCREP`, and `PCREP`" [483]

- Growth factors report: Section 10.2.2, "`APROJREP`, `MPROJREP`, and `PPROJREP`" [484]

- Reactivity controls report: Section 10.2.3, "`AREACREP`, `MREACREP`, and `PREACREP`" [484]

- Summary of controls report: Section 10.2.4, "`ACSUMREP`, `MCSUMREP`, and `PCSUMREP`" [485]

10.2.1. ACREP, MCREP, and PCREP

Type: ASCII - list-directed

Summary: This file contains the report for the multiplicative control matrix. It includes the inventory pollutant emissions before and after application of the multiplicative control matrix factors. For area and mobile sources, the report resolution is by state and SCC code. For point sources, the report is by state/county and plant code. The individual factors applied (e.g., /ALLOWABLE/ and /CONTROL/ packet factors) by

source are provided in the *CSUMREP file described in Section 10.2.4, "ACSUMREP, MCSUMREP, and PCSUMREP" [485].

The file header and output format are similar to what is output by the **Smkreport** program (see Section 7.4, "Reports Created by **Smkreport**" [326]. The headers include information about:

*   The source category (area, mobile, or point sources) for the report
*   The base year or projected year of the inventory input file
*   Which control packets were applied
*   Whether the emissions in the report are based on ozone-season or annual total emissions
*   The inventory pollutants controlled by the packets
*   The units

The columns of the report include:

*   State ID (area and mobile); state/county ID (point)
*   SCC (area and mobile); plant (point)
*   Input emissions in tons/day for each controlled pollutant
*   Output emissions in tons/day for each controlled pollutant

10.2.2. APROJREP, MPROJREP, and PPROJREP

Type: ASCII - list-directed

Summary: This file contains the report for the /PROJECTION/ packet applied during the **Cntlmat** run. It includes the growth matrix factors applied to the inventory pollutant emissions. For all sources, the resolution of the report is the source level. For area and mobile sources, this means that the report resolution is by county code and SCC. Mobile sources may additionally have link-detail information if links are included in the inventory. For point sources, the resolution includes the state/county code, plant ID, SCC, and other source identifiers such as segment and point.

The file header and output format are somewhat similar to that output by the **Smkreport** program (see Section 7.4, "Reports Created by **Smkreport**" [326]), but less so than the *CREP file (see Section 10.2.1, "ACREP, MCREP, and PCREP" [483]). The headers include information about:

*   The source category (area, mobile, or point sources) for the report
*   The base year of the inventory
*   The growth year for the packets being applied
*   The fact that all pollutants receive factors uniformly.

The columns of the report include:

*   State ID (area and mobile); state/county ID (point)
*   SCC (area and mobile); plant (point)
*   Growth factor

10.2.3. AREACREP, MREACREP, and PREACREP

Type: ASCII - list-directed

Summary: This file contains the report for the reactivity packet applied during the **Cntlmat** run. It contains the reactivity packet contents applied for any source in the inventory that gets a reactivity control. For all sources, the resolution of the report is the source level, as described for the [A|M|P]PROJREP files in Section 10.2.2, "APROJREP, MPROJREP, and PPROJREP" [484].

The file header and output format are the same as that output by the **Smkreport** program (see Section 7.4, "Reports Created by **Smkreport**" [326]). The headers in include information about:

- The source category (area, mobile, or point sources) for the report
- The base year of the inventory (or the projected year if applicable)
- The projected year in the reactivity packet
- The pollutant name (e.g., ROG) for which the reactivity controls were applied.
- Whether the emissions in the report are based on average-day or annual total emissions.
- The column names and units

The columns in the report include:

- SMOKE source number
- State/county code
- Base-year SCC
- Base-year emissions
- New base-year emissions
- Projection factor
- Projected-year SCC
- Projected-year speciation profile
- Market penetration rate

10.2.4. `ACSUMREP`, `MCSUMREP`, and `PCSUMREP`

Type: ASCII - list-directed

Summary: This file contains all of the control (not growth) information applied to each source, by source. The structure of this file is not amenable for import into a spreadsheet, because all control information applied to each source is listed underneath each source (not on the same line). There is no header in this file.

10.3. Elevpoint

The **Elevpoint** program is responsible for selecting the elevated sources. **Elevpoint** creates the following output files:

- Elevated-point-source report: Section 10.3.1, "`REPPELV`" [485]

- Stack groups file: Section 10.3.2, "`STACK_GROUPS`" [486]

10.3.1. `REPPELV`

Type: ASCII - list-directed

Summary: This ASCII output file contains the sources and criteria used for selecting elevated and PinG sources in SMOKE. The following is a description of columns/header in the report:

- Source ID: SMOKE source ID, used by SMOKE only

- Region: Country/state/county code (CSSYYY where C=country, SS=state, and YYY=county)

- Plant: Inventory facility ID

- Char 1: For IDA inventory, Point ID

- Char 2: For IDA inventory, Stack ID

- Char 3: For IDA inventory, Segment ID

- Char 4: For IDA inventory, SCC

- Plt Name: Plant description from the input inventory

- Elevstat: Elevated status - P = plume-in-grid, E = elevated

- Group: SMOKE-assigned stack group number

- Stk Ht: Stack height in meters

- Stk Dm: Stack diameter in meters

- Stk Tmp: Stack exit gas temperature in K

- Stk Vel: Stack exit gas velocity in m/s

- Stk Flw: Stack exit gas flow rate in m^3/s

- Rise: Briggs analytical solution plume rise in meters. This column is output only when RISE is specified as a selection criterion in the `PELVCONFIG` input file.

- Group <pol>: Daily maximum emissions for source or source group for each pollutant (<pol>) in the `PELVCONFIG` file. These emissions are source emissions when a single source has its own group number. This column is output only when a pollutant is specified as a selection criterion in the `PEL-VCONFIG` input file.

- Var 1: The first variable from the `PELVCONFIG` input file that was used in establishing the source as a PinG/elevated source

- Type 1: If "TOP" was used in the `PELVCONFIG` input file, this field says "RANK"

- Test 1: The test used to establish this source as a PinG/elevated source (e.g., >, <, >=, =)

- Val 1: If TOP was used, this value is the RANK. Otherwise, this value is the criterion used with Var 1 and Test 1 (e.g., if Var 1 is HT, Test 1 is ">", and if Val 1 is 50, then the criterion for selection is that the stack height be greater than 50.

Var 2 through Val 2 are the same as Var 1 through Val 1, except these are defined only if an AND condition existed for the OR condition that SMOKE used to establish the source as PinG or elevated. There will be as many variables of these types as there are AND conditions in the **PELVCONFIG** file.

10.3.2. `STACK_GROUPS`

Type: I/O API NetCDF - adapted gridded format

Summary: I/O API intermediate file and output file for CMAQ. See Section 9.3.2, "`STACK_GROUPS`" [464] for more information.

10.4. Mrggrid

The **Mrggrid** program is responsible for creating model-ready emissions files. **Mrggrid** creates the following output file:

- Model-ready emissions file: Section 10.4.1, "`OUTFILE`" [487]

10.4.1. `OUTFILE`

Type: I/O API NetCDF - adapted gridded format

Summary: Final, model-ready file created from two or more souce categories. See Section 10.6, "**Smkmerge**" [488] for file format.

10.5. Smkinven

The **Smkinven** program is responsible for importing the raw inventory into SMOKE format. **Smkinven** creates the following output files:

- Inventory import report: Section 10.5.1, "`REPINVEN`" [487]

- SCC file: Section 10.5.2, "`ASCC`, `MSCC`, `PSCC`" [488]

10.5.1. `REPINVEN`

Type: ASCII - list-directed

Summary: This report is made up of several different sections describing different parts of the inventory import.

1. This section lists, by CAS number, the emissions in tons/year; the number of inventory records; whether all, some, or none of the pollutants associated with that CAS number were kept; and the CAS number description.

   The columns in the report include:
   - CAS number
   - Whether all, some, or none of the pollutants were kept
   - Number of inventory records read for the CAS number
   - Emissions for the CAS number
   - CAS number description

2. This section includes emissions before and after application of disaggregation factors from the `INVT-ABLE`. Only the pollutants that have Keep = Y in `INVTABLE` will be included in this report. This report will be provided only if disaggregation and/or aggregation have occurred.

   The columns in the report include:
   - "Kept" unique CAS number
   - Emissions total for the CAS number
   - Factor used to adjust the CAS number to the pollutant
   - Pollutant name
   - Emissions total of the inventory pollutant
   - Pollutant description

• CAS number description

The following sections are only for area-to-point assignments:

3. This section lists SCCs that have area-to-point source factor file assignments but are not in the inventory.

The columns in the report include:

• Assigned SCC codes not in the inventory

4. For emissions going to point sources, this section lists the pollutant name, emissions total before and after the factors are applied, and the total number of FIPS codes affected, by SCC.

The columns in the report include:
• SCC code
• Pollutant name
• Number of FIPS codes affected
• Emissions before factors are applied
• Emissions after factors are applied
• SCC description

5. Same as the above section, except emissions are reported by state.

The columns in the report include:
• State
• SCC code
• Pollutant name
• Number of FIPS codes affected
• Emissions before factors are applied
• Emissions after factors are applied
• SCC description

6. This section lists, by SCC, the number of state and county FIPS codes getting assigned to point locations and the number not getting assigned.

The columns in the report include:
• SCC code
• Number of FIPS assigned to point locations
• Number of FIPS not assigned to point locations
• SCC description

10.5.2. `ASCC, MSCC, PSCC`

Type: ASCII - list-directed

Summary: This file simply lists the SCC codes in the raw inventory.

10.6. Smkmerge

The **Smkmerge** program is responsible for creating model-ready emissions files. **Smkmerge** creates the following output file:

• Model-ready emissions file: Section 10.6.1, "`[A|B|M|P|E]G[T][S][3D][_S][_L]`" [489]

10.6.1. `[A|B|M|P|E]G[T][S][3D][_S][_L]`

Type: I/O API NetCDF - adapted gridded format

Table 10.1. I/O API Header for `[A|B|M|P|E]G[T][S][3D][_S][_L]`

I/O API Header variable	Meaning	Set by...
NCOLS3D	Number of columns in the grid	Output grid description in GRIDDESC
NROWS3D	Number of rows in the grid	Output grid description in GRIDDESC
NVARS3D	Number of output species	Chemical mechanism and speciation profiles in GSPRO

Table 10.2. Variables in `[A|B|M|P|E]G[T][S][3D][_S][_L]`

Variable Name	Type	Units	Description
Name of species 1	Real	[moles|tons]/time	Emissions for species 1
...	...	...	...
Name of species NVARS3D	Real	[moles|tons]/time	Emissions for species NVARS3D

10.7. Tmpbeis3

The **Tmpbeis3** program is responsible for creating gridded, hourly biogenic emissions. **Tmpbeis3** creates the following output file:

- Model-ready biogenic emissions: Section 10.7.1, "B3GTS_L and B3GTS_S" [489]

10.7.1. `B3GTS_L and B3GTS_S`

Type: I/O API NetCDF - adapted gridded format

Table 10.3. I/O API Header for `B3GTS_L and B3GTS_S`

I/O API Header variable	Meaning	Set by...
NCOLS3D	Number of columns in the grid	Output grid description in GRIDDESC
NROWS3D	Number of rows in the grid	Output grid description in GRIDDESC
NVARS3D	Number of output species	Chemical mechanism and speciation profiles in GSPRO

Table 10.4. Variables in `B3GTS_L and B3GTS_S`

Variable Name	Type	Units	Description
Name of species 1	Real	[moles|tons]/hr	Emissions for species 1
...	...	...	...
Name of species NVARS3D	Real	[moles|tons]/hr	Emissions for species NVARS3D

10.8. Tmpbio

The **Tmpbio** program is responsible for creating gridded, hourly biogenic emissions. **Tmpbio** creates the following output file:

- Model-ready biogenic emissions: Section 10.8.1, "`BGTS_L` and `BGTS_S`" [490]

10.8.1. `BGTS_L` and `BGTS_S`

Type: I/O API NetCDF - adapted gridded format

The `BGTS_L` and `BGTS_S` files have the same format as the `B3GTS_L` and `B3GTS_S` files, respectively. See Section 10.7.1, "`B3GTS_L` and `B3GTS_S`" [489] for format information.

Chapter 11. Source Code and Include Files

Table of Contents

The SMOKE modeling system consists of several core programs (described in Chapter 6, *SMOKE Core Programs* [211]), several utility programs (described in Chapter 5, *SMOKE Utility Tools* [177], and the quality assurance program **Smkreport** (described in Chapter 7, *SMOKE Quality Assurance* [305]). The source code for all programs is available in the `$SMKROOT/src` directory.

SMOKE also uses several libraries for shared functionality. The I/O API and NetCDF libraries were discussed in Chapter 12, *Downloading, Installing, and Compiling SMOKE* [495]. In addition, SMOKE is distributed with the FileSetAPI library (Section 11.1.1, "FileSetAPI library" [491]), and the EDSS Tools library (Section 11.1.2, "EDSS Tools library" [491]).

11.1. Source Code

All source code for SMOKE and its associated libraries can be found in the `$SMK_SUBSYS` directory.

11.1.1. FileSetAPI library

This library is used to open multiple I/O API files to avoid the limit of 120 variables per I/O API file. The source code in found in the `$SMK_SUBSYS/filesetapi` directory (also accessible using the `FS_ROOT` environment variable), which contains the following source code and Makefiles.

- `$FS_ROOT/appendname.f`
- `$FS_ROOT/chkfileset.f`
- `$FS_ROOT/chksetdesc.f`
- `$FS_ROOT/cleanup.f`
- `$FS_ROOT/closeset.f`
- `$FS_ROOT/createset.f`
- `$FS_ROOT/descset.f`
- `$FS_ROOT/Makefile`
- `$FS_ROOT/Makeit`
- `$FS_ROOT/modfileset.f`
- `$FS_ROOT/modmake.inc`
- `$FS_ROOT/openset.f`
- `$FS_ROOT/promptset.f`
- `$FS_ROOT/readset.F`
- `$FS_ROOT/SETDECL.EXT`
- `$FS_ROOT/writeset.F`

11.1.2. EDSS Tools library

The EDSS Tools library is built on the EDSS framework and contains a variety of routines used for accessing I/O API data and other Models-3 conventions. The main directory for the EDSS Tools library is

$SMK_SUBSYS/edss_tools, which is also accessible with the TOOLS_ROOT environment variable. The source code for the EDSS Tools library is in the $TOOLS_ROOT/src directory and consists of the following source code and Makefiles.

- $ETINC/IOCNST3.EXT
- $ETINC/IOPRVT3.EXT
- $ETINC/IOSTRG3.EXT
- $ETLSRC/checkmem.f
- $ETLSRC/chkgrid.f
- $ETLSRC/chkint.f
- $ETLSRC/chkreal.f
- $ETLSRC/dscm3grd.f
- $ETLSRC/dscm3lay.f
- $ETLSRC/flterr.f
- $ETLSRC/fltrneg.f
- $ETLSRC/getcfdsc.f
- $ETLSRC/getfline.f
- $ETLSRC/getifdsc.f
- $ETLSRC/getm3epi.f
- $ETLSRC/getnlist.f
- $ETLSRC/hdrmiss3.f
- $ETLSRC/ioapi_grd_size.f
- $ETLSRC/m3mesg.F
- $ETLSRC/Makefile
- $ETLSRC/Makeit
- $ETLSRC/multunit.f
- $ETLSRC/nextime.f
- $ETLSRC/openphys.f
- $ETLSRC/padnzero.f
- $ETLSRC/padzero.f
- $ETLSRC/parsline.f
- $ETLSRC/polmesg.f
- $ETLSRC/rd3mask.f
- $ETLSRC/rdsetmask.f
- $ETLSRC/unitfac.f
- $ETLSRC/unitmatch.f
- $ETDSRC/modgrid.f
- $ETDSRC/modmake.inc

11.1.3. SMOKE programs

The source code needed for the SMOKE programs is in the $SMKROOT/src directory. Table 11.1, "SMOKE programs and source code directories" [493] lists each SMOKE program and the corresponding source code directory and environment variable you can use to access that directory.

Table 11.1. SMOKE programs and source code directories

SMOKE Program	Source Code Directory	Environment Variable
Cntlmat	`$SMKROOT/src/cntlmat`	`CL_SRC`
Elevpoint	`$SMKROOT/src/point`	`PT_SRC`
Geofac	`$SMKROOT/src/emutil`	`UT_SRC`
Grdmat	`$SMKROOT/src/grdmat`	`GD_SRC`
Grwinven	`$SMKROOT/src/smkinven`	`IV_SRC`
Invsplit	`$SMKROOT/src/emutil`	`UT_SRC`
Laypoint	`$SMKROOT/src/point`	`PT_SRC`
Metcombine	`$SMKROOT/src/emutil`	`UT_SRC`
Metscan	`$SMKROOT/src/emutil`	`UT_SRC`
Met4moves	`$SMKROOT/src/movesmrg`	`MV_SRC`
Movesmrg	`$SMKROOT/src/movesmrg`	`MV_SRC`
Mrggrid	`$SMKROOT/src/smkmerge`	`MG_SRC`
Normbeis3	`$SMKROOT/src/biog`	`BG_SRC`
Pktreduc	`$SMKROOT/src/emutil`	`UT_SRC`
Smk2emis	`$SMKROOT/src/emutil`	`UT_SRC`
Smkinven	`$SMKROOT/src/smkinven`	`IV_SRC`
Smkmerge	`$SMKROOT/src/smkmerge`	`MG_SRC`
Smkreport	`$SMKROOT/src/emqa`	`QA_SRC`
Spcmat	`$SMKROOT/src/spcmat`	`SP_SRC`
Surgtool	`$SMKROOT/src/emutil`	`UT_SRC`
Temporal	`$SMKROOT/src/temporal`	`TM_SRC`
Tmpbeis3	`$SMKROOT/src/biog`	`BG_SRC`
Uam2ncf	`$SMKROOT/src/emutil`	`UT_SRC`

There are three additional directories in the `$SMKROOT/src` that we have not mentioned. They are:

- `$SMKROOT/src/emmod` (`MD_SRC`): Contains modules for shared program variables

- `$SMKROOT/src/inc` (`INC`): Contains include files (discussed in Section 11.2, "Include Files" [493])

- `$SMKROOT/src/lib` (`EL_SRC`): Contains source code for the main SMOKE library

11.2. Include Files

In this section we list all the include files used to compile the various libraries and SMOKE programs.

11.2.1. I/O API

These include files are part of the I/O API library and are in the `$SMK_SUBSYS/ioapi_includes` directory, accessible using the environment variable `IOINC`.

- `$IOINC/ATDSC3.EXT`

- `$IOINC/CONST3.EXT`
- `$IOINC/FDESC3.EXT`
- `$IOINC/IODECL3.EXT`
- `$IOINC/NETCDF.EXT`
- `$IOINC/NOTICE.EXT`
- `$IOINC/PARMS3.EXT`
- `$IOINC/STATE3.EXT`

11.2.2. FileSetAPI

- `$FS_ROOT/SETDECL.EXT`

11.2.3. EDSS Tools

- `$ETINC/IOCNST3.EXT`
- `$ETINC/IOPRVT3.EXT`
- `$ETINC/IOSTRG3.EXT`

11.2.4. SMOKE

- `$CL_SRC/CPKTDAT.EXT`
- `$INC/B3DIMS3.EXT`
- `$INC/B3V12DIMS3.EXT`
- `$INC/BIODIMS3.EXT`
- `$INC/EMCNST3.EXT`
- `$INC/EMPRVT3.EXT`
- `$INC/EMSTRG3.EXT`
- `$INC/FLTERR.EXT`
- `$INC/M6CNST3.EXT`

Chapter 12. Downloading, Installing, and Compiling SMOKE

Table of Contents

12.1. Introduction

Instructions on how to download and install SMOKE are provided in Section 12.2, "Downloading SMOKE" [495] and Section 12.3, "Installing SMOKE" [495]. Section 12.4, "Compiling SMOKE for UNIX" [496] describes how to compile SMOKE for UNIX platforms other than those supported by CMAS. The SMOKE compile scripts are configured to use a UNIX C-shell environment, so they are not useful for creating Windows-based SMOKE files; using SMOKE with any version of the Windows operating system is not supported. If your operating system is one of the types supported by CMAS, you need only to download the SMOKE executables available from the CMAS Model Clearinghouse [http://www.cmascenter.org/html/models.html].

12.2. Downloading SMOKE

Steps for downloading SMOKE:

1. Go to the CMAS Model Clearinghouse [http://www.cmascenter.org/html/models.html].

2. Click on the SMOKE link in the list of CMAS-supported products.

3. If you have already registered with CMAS, enter your email address and password. Otherwise, click the "Register now" link and complete the registration form.

4. You will be sent to the CMAS Download Center. Select SMOKE as the product you wish to download and select your computer and compiler type.

5. Download the installation script (`smoke_install.csh`), the source code, scripts, and executables archive (`smoke_${version}.Linux2_x86_64pg.tar.gz`), and the test case data files (`smoke_${version}.nctox.MOVES.data.tar.gz`). Save these files in the same directory on your machine.

12.3. Installing SMOKE

Steps for installing SMOKE:

1. Make sure that the three files you downloaded from the CMAS website are in the same directory.

2. Choose the directory where you want to install SMOKE and the data files. This directory will be known as your `SMK_HOME` directory.

3. At the command line, type:

    setenv SMK_HOME *<your selected directory for SMOKE installation>*

    NOTE: the SMOKE scripts are designed to run under the UNIX C-shell (either **csh** or **tcsh**); the scripts will not work with other shells such as **bash**.

4. Make sure your SMK_HOME directory exists by typing the following command on the command line:

    mkdir -p $SMK_HOME

5. Go to the directory when you have saved the SMOKE installation files.

6. Type the following at the command line:

    source smoke_install.csh

    This command will unzip and install the SMOKE files in your SMK_HOME directory. If the script encounters any problems, it will print an error message to point you to the problem.

The SMK_HOME setting is needed by the SMOKE scripts. Therefore, it is best to always have this setting defined when you log into your computer. To do this, perform the following additional steps before proceeding:

7. Go to your home directory, then use a text editor to view the .cshrc file (note that this file is a hidden file, which can be listed in UNIX using the **ls -a** command at the prompt).

8. In the .cshrc file, add the following line:

    setenv SMK_HOME *<your selected directory for SMOKE installation>*

9. Save and close the .cshrc file. Your SMK_HOME directory will now be defined automatically when you log into the computer or when you open a new **xterm** window.

Now you are ready to run the SMOKE nctox case. Running SMOKE is described in Chapter 4, *Using SMOKE Scripts* [117], which discusses using the SMOKE scripts. Please refer to Section 4.3, "Running the SMOKE test case" [124] for instructions on running the example scripts for the nctox case. The current SMOKE documentation can be found online at http://www.smoke-model.org.

12.4. Compiling SMOKE for UNIX

If the executables for your platform are not available from the CMAS web site, you will need to compile SMOKE for your system. You may also need to compile SMOKE if you would like to modify the source code. The following steps will help get you set up to compile SMOKE.

*   Install SMOKE as described in Section 12.3, "Installing SMOKE" [495].

*   Download and compile the I/O API library from the I/O API home page [https://www.cmascenter.org/ioapi/]. The I/O API can be installed in $SMK_HOME/subsys/ioapi

    [1] Create symbolic link to the IOAPI source code directory (ioapi) to the $SMK_HOME/subsys/ioapi/ioapi/ directory.

    [2] Create symbolic link to the IOAPI libioapi.a library file and utility programs (datshift, jul2greg, greg2jul and so on) to the $SMK_HOME/subsys/ioapi/Linux2_x86_64ifort/ directory.

- After the appropriate libraries and compile options have been installed, you can source an Assigns file to ensure that your system is configured properly.

  `cd $SMK_HOME/subsys/smoke/assigns`

  `source ASSIGNS.nctox.cmaq.cb05_soa.us12-nc`

- You should now be ready to compile the SMOKE programs. Change directories to `$SMKOOT/src/`.

- At the UNIX prompt, type:

  `make`

- After running the `Makeall` script, you should check the `$SMK_BIN` directory to check the executables.

- Once you have compiled the SMOKE programs, you can run the SMOKE example scripts following the instructions in Section 4.3, "Running the SMOKE test case" [124].

Appendix A. Glossary

A

Air quality model

A model that predicts ambient air quality that accounts for impacts of meteorology, manmade emissions sources, and biogenic emissions sources. It simulates physical components such as advection, dispersion, clouds, dry and wet precipitation, air chemistry, particulate formation, to provide ambient air quality estimates of a variety of pollutants (such as ozone) and other chemical constituents.

Anthropogenic sources

Manmade sources of emissions. All inventory source categories besides biogenic sources are considered to be anthropogenic sources.

Area source

An anthropogenic source category that is represented as a 2-d source, typically a county. In SMOKE, the term area source more specifically refers to a collection of source categories types that are not point sources and are not on-road mobile sources. SMOKE area sources are defined by country, state, and county codes and by source category codes (SCCs). Area sources are also referred to as stationary area sources and, in SMOKE, include nonroad mobile sources.

B

Base year

Year for which an air quality modeling episode will be evaluated for model performance. Air quality modeling typically involves modeling a base year and comparing the model run to observational data. Once the meteorology and emissions data in the base year have been im-proved enough for adequate model performance, the emissions are grown to a future year and the model is run in the future with and without a control strategy.

Biogenic source

A natural source, or one that is not manmade. In SMOKE, biogenic sources include VOC emissions from vegetation and nitrous oxide emissions from soil. Volcanoes and other geothermal emissions, water emissions, and other biogenic sources are not included in SMOKE biogenic emissions components.

C

Chemical mechanism

A set of chemical species and their interactions used to represent air chemistry (e.g., carbon bond 4 [CB4], Regional Acid Deposition Model [RADM], Statewide Air Pollution Research Center [SAPRC]).

Chemical speciation

Convert the inventory pollutant data to the chemical species needed by the air quality model (e.g., VOC gets split to PAR, OLE, XYL, TOL, ISOP, and more).

Cross-reference file

A dataset used for matching sources in the emissions inventory with profile data based on the source characteristics. Typically there can be thousands of records in a cross-reference file and tens or hundreds

of thousands of records in an emissions inventory. The cross-reference file facilitates assigning conversion factors for emissions processing to records in the emissions inventory.

E

Elevated source

A point source in which emissions are higher than the first model layer because of plume rise.

Emission inventory

An emission inventory is an accounting of the amount of air pollutants discharged into the atmosphere. It is generally characterized by the following factors:

- the chemical or physical identity of the pollutants included,
- the geographic area covered,
- the institutional entities covered,
- the time period over which emissions are estimated, and
- the types of activities that cause emissions.

Emission inventories are developed for a variety of purposes. Inventories of natural and anthropogenic emissions are used by scientists as inputs to air quality models, by policy makers to develop strategies and policies or track progress of standards, and by facilities and regulatory agencies to establish compliance records with allowable emission rates. A well constructed inventory should include enough documentation and other data to allow readers to understand the underlying assumptions and to reconstruct the calculations for each of the estimates included.

Emissions processing

Converting the resolution of emissions inventories (e.g. area, biogenic, mobile, and point sources) to the resolution needed by an air quality model (e.g., hourly, grid-cell, data of model species). It may also include growing data from one year to another year and applying emissions controls on the inventory. This process requires a large number of supplementary data files include gridding surrogates, cross-reference files, and factor profiles for chemical speciation and temporal allocation. It includes combining all source categories and ensuring that the emissions are in the format needed by a specific air quality model. Emissions processing typically includes quality assurance to increase user confidence that the data were converted as intended.

Environment variable

Environment variables are how SMOKE communicates with its operating environment. There are several different uses for environment variables including assigning file names, setting options, and setting operating parameters.

An environment variable in the UNIX environment C-shell is defined using the **setenv** command. For example, to define the `FILENAME` environment variable to the file `testfile.txt` in the `/home/user/` directory, the following command is used:

`setenv FILENAME /home/user/testfile.txt`

F

Fuel month

The concept of a fuel month is used to indicate when a particular set of fuel properties should be used in a MOVES simulation. The fuel month reduces the computational time of MOVES by using a single month to represent a set of months. To determine the fuel month and which months it is similar to, the user should review the State-provided fuel supply data in the MOVES database for each representative county. If the fuel supply data change thoughout the year, then roup the months by fuel parameters. For example, if the grams/mile exhaust emission rates in January are identical to February's rates for a given representative county, then use a single fuel month to represent January and Febuary. (Only one of the months needs to be modeled throuh MOVES)

Future year

A year after the base year.

G

Grid

See Model grid.

Grid projection

See Map projection.

Gridding

See Spatial allocation.

Gridding surrogate

A dataset developed from data at a finer resolution than the emissions, used to spatially allocate the emissions to the grid cells (e.g., population, housing, agricultural regions).

I

Inventory pollutant

A compound or group of compounds emitted into the atmosphere by anthropogenic and biogenic sources defined for record-keeping and regulatory purposes (e.g. carbon monoxide [CO], nitrogen oxides [NO_x], volatile organic compounds [VOC], particulate matter 10 microns or less [PM_{10}]).

L

Lambert conformal

A type of map projection.

Latitude/longitude (lat/lon)

A spherical reference system used to measure locations on the Earth's surface. Latitude and longitude are angles measured from the Earth's center to locations on the Earth's surface. Latitude measures angles in a north-south direction. Longitude measures angles in the east-west direction. In SMOKE, lat/lon refers to a map projection.

Layer

See Model layers.

Logical file names

Environment variables that programs use to access files are called logical file names. In the Environment variable [500], `FILENAME` is the logical file name for the physical file `/home/user/test-file.txt`. The I/O API is based on logical file names, and since

SMOKE uses the I/O API for accessing all of its files, it too uses logical file names. The benefit of logical file names is that the programs do not require that the physical files always have the same name. Instead, the logical file names can be defined each time a program is run to use whatever physical file names the user would like. In Chapter 8, *SMOKE Input Files* [345], Chapter 9, *SMOKE Intermediate Files* [461], and Chapter 10, *SMOKE Output Files* [483], the logical file names are used to reference the files, their associations with programs, and their formats.

Low-level point source | A point source in the first model layer. A source that is not elevated.

M

Map projection | A mathematical model that transforms the locations of features on the Earth's surface to locations on a two-dimensional surface. Because the Earth is three-dimensional, some method must be used to depict a map in two dimensions. Some projections preserve shape; others pre-serve accuracy of area, distance, or direction. Map projections project the Earth's surface onto a flat plane. However, any such representation distorts some parameter of the Earth's surface be it distance, area, shape, or direction.

Merging | In SMOKE, merging can refer to two steps. One, it can refer to com-bining the SMOKE processes for a single source category, such as combining the gridding information, the inventory, the temporal alloc-ation, and the chemical speciation information to create model-ready emissions. Two, it can refer to combining all source categories together into a single model-ready file.

Meteorology data | Simulated weather data required by SMOKE for biogenic sources, some mobile source processing, and some point source processing.

Mobile source | In the most general sense, a mobile source is simply a non-stationary source such as an on road vehicle, a construction vehicle, or a lawn mower. In SMOKE, the term mobile source more specifically refers to on-road vehicles. SMOKE mobile sources are defined by country, state, and county codes, by source category codes (SCCs) that include a road class code and a vehicle type code, and optionally by a link identification code. For SMOKE, mobile sources include only on-road mobile sources and NOT nonroad sources (which are treated as area sources).

Model domain | The grid on which a model will be run.

Model grid | A 2-d grid based on a map projection, defined by starting coordinates, number of grid cells in each direction, and the physical size of the grid cells.

Model layers | Vertical spatial divisions defined by an air quality model because the atmosphere has varied characteristics in the vertical direction.

Model species | A compound or group of compounds defined as part of the estimation of air chemistry in an air quality model (e.g., CO, nitrogen oxide [NO],

nitrogen dioxide [NO$_2$], paraffins [PAR], elemental carbon, coarse [ECC]). Model species can be an actual chemical compound, ion, or group of compounds. Air quality models typically input emissions data for a subset of model species from a larger number of species in a full chemical mechanism.

N

Nonroad mobile sources	See Area source.

O

On-road mobile source	See Mobile source.

P

Past year	A year prior to a base year.
Plume rise	The rising of the exhaust from point sources due to the velocity and temperature of the exhaust gases.
Plume-in-grid (PinG)	A special treatment of elevated sources by which the plume rise is modeled with extra detail by the air quality model. The acronym for plume-in-grid is PinG or sometimes PIG.
Point source	A point source is an emissions source that can be represented by a point in space, relative to the modeling domain. In SMOKE, a point source is defined by a country, state, and county code, a plant identification code, 1 to 5 additional user-defined attributed (such as stack code, process codes, and a source category code [SCC]). Point sources have other required attributes such as stack parameters.
Polar stereographic	A type of map projection.
Profile data	Factors used for disaggregating emissions data as is done during chemical speciation or temporal allocation.
Projection (grid)	See Map projection.
Projection (temporal)	Estimating past-year or future-year emissions from a base-year value and other factors.

R

Reference County	Reference county is used to represent a set of counties that have the a constant set of fuel parameters (distribution of fuels over the year), fleet age distritution, inspection/maintenance (I/M) programs, similar community characteristics (e.g. urban vs rural) and similar meteorological conditions (e.g. temperature and relative humidity). Variables that can vary within the counties represented by a representative county include vehicle miles traveled (VMT), source type vehicle population, roadway speed and grid cell temperatures. By specifying a represent-

ative county the user is able to reduce the computational time of MOVES by using a single county to represent a set of counties.

S

Source category	A general classification of emission sources for both emission inventory development and emissions modeling. The major source categories in SMOKE are area, biogenic, mobile, and point. Other source categories that are sometimes treated separately are offshore point sources, non-road area sources, and Continuous Emissions Monitoring (CEM) sources. See Also Area source, Biogenic source, Mobile source, Point source.
Source classification code (SCC)	A code used in emission inventories to label emission sources as being part of a specific industrial, chemical, or natural processes of generating emissions.
Sparse Matrix Operator Kernel Emissions (SMOKE)	A emissions data modeling system that prepares emissions data for use in air quality models. It converts the resolution of area, biogenic, mobile, and point source inventories into the hourly, gridded, model-species resolution needed by air quality models.
Spatial allocation	Convert the source spatial extent to the grid cell resolution needed by the air quality model. For area sources, this process requires gridding surrogate data. For mobile link sources (line sources), this requires determining the cells intersected by the link and the fractions of the link in each cell. For point sources, this requires determining in which cell the point source falls.
Spatial surrogate	See Gridding surrogate.
Speciation	See Chemical speciation.
Species	See Model species.
Stack parameters	Attributes of point sources including stack height, stack diameter at the top (the point where exit gases flow from the stack), exit gas temperature at the top, exit gas velocity at the top, and exit gas flow at the top.

T

Temporal allocation	Convert the inventory temporal resolution to the hourly temporal resolution needed by the air quality model.
Time zone	A region on the Earth assigned a specific zone relative to Greenwich Mean Time, defined as time zone 0.

U

Universal Transverse Mercator	Military grid projection based on the transverse Mercator projection, applied to maps of the Earth's surface extending from the Equator to 84 degrees north and 80 degress south latitudes.

UNIX	A group of operating systems that conform to POSIX standards. Commonly used systems include Linux, SGI IRIX, Solaris (SunOS), and Mac OS X.
UTM zone	See Universal Transverse Mercator.

Appendix B. Abbreviations

Abbreviation	Expansion
AMS	area and mobile source
AQM	air quality model
ATP	antitampering program
BEIS3	Biogenic Emission Inventory System, version 3
BELD4	Biogenic Emission Landcover Database, version 3.60
BELD3	Biogenic Emission Landcover Database, version 3.14
BRK	particulate matter from brake component wear
CAA	Clean Air Act
CAM_X	Comprehensive Air Quality Model, with Extensions
CB4	Carbon Bond 4
CEI	crankcase extended idle exhaust
CEP	Carolina Environmental Program
CEM	Continuous Emissions Monitoring
CHIEF	Clearinghouse for Inventories and Emission Factors
CMAQ	Community Multiscale Air Quality
CMAS	Community Modeling and Analysis System
CO	carbon monoxide
CRC	evaporative crankcase emissions (blow-by)
CTG	control technology guideline
CXS	crankcase start exhaust
DDDYYYY	date in Julian format (3-digit day plus 4-digit year)
DNL	evaporative diurnal emissions (heat rise)
EDSS	Environment Decision Support System
EFL	evaporative fuel leaks
EFV	evaporative fuel vapor venting
EGU	electric generating utility
EMC	Environmental Modeling Center
EMS-95	Emissions Modeling System, '95
EPA	U.S. Environmental Protection Agency
EPM	Evaporative Permeation
EPS	Emissions Processor System
EXR	exhaust running emissions
EXS	exhaust engine start emissions (trip start)
EXT	extended idle exhaust
FF10	Flat File 10
FIA	Forest Inventory Area
FIPS	U.S. Federal Implementation Planning Standards
GIS	geographic information system
GMT	Greenwich Mean Time
GUI	graphical user interface
HAP	hazardous air pollutant
HHMMSS	hours (2 digits), minutes (2 digits), seconds (2 digits); other forms are HH-HMMSS and HHHHMMSS, which differ only in the number of digits in the hours position
HPC	high-performance computing
HOT	evaporative hot soak emissions (trip end)
IDA	Inventory Data Analyzer
I/M	inspection and maintenance
I/O	input/output
I/O API	Input/Output Applications Programming Interface

Abbreviation	Expansion
lat-lon	latitude-longitude
LMOS	Lake Michigan Ozone Study
MACT	Maximum Achievable Control Technology
MAQSIP	Multiscale Air Quality Simulation Platform
MCIP	Meteorology-Chemistry Interface Processor
MCPL	Meteorology Coupler
MIMS	Multimedia Integrated Modeling System
MM5	Pennsylvania State University/National Center for Atmospheric Research fifth-generation Mesoscale Model
MPS	major point source
MEPSE	major elevated point-source emissions (same as PinG)
MMDDYYYY	date in Gregorian format (2-digit month plus 2-digit day plus 4-digit year)
MOVES	Motor Vehicle Emission Simulator
MWSS	Monday-weekday-Saturday-Sunday
NAICS	North American Industrial Classification System
NC DENR	North Carolina Department of Environment and Natural Resources
NEI	National Emissions Inventory
NetCDF	Network Common Data Form
NH3	ammonia
NonEGU	Non-electric generating utility
NOx	nitrogen oxides
NTI	National Toxics Inventory
OAQPS	Office of Air Quality Planning and Standards
ORD	Office of Research and Development
ORIS	Office of Regulatory Information Systems
ORL	one record per line
OTAG	Ozone Transport and Assessment Group
PAR	photosynthetic active radiation
PAH	polynuclear aromatic hydrocarbons
PinG	plume in grid (same as MEPSE)
PM	particulate matter
PM10	particulate matter 10 micrometers or smaller in diameter
PM2.5	particulate matter 2.5 micrometers or smaller in diameter
PMC	particulate matter coarse
QA	quality assurance
RACT	reasonably available control technology
RADM	Regional Acid Deposition Model
REMSAD	Regulatory Modeling System for Aerosols and Deposition
RFG	reformulated gas
RFL	evaporative refueling emissions (fuel displacement and spillage)
RH	Relative Humidity
RMC	Regional Modeling Center
RPD	Rate Per Distance
RPP	Rate Per Profile
RPV	Rate Per Vehicle
RST	evaporative running loss emissions
SAPRC	Statewide Air Pollution Research Center
SCC	Source Classification Code
SIC	Standard Industrial Classification
SIP	State Implementation Plan
SMOKE	Sparse Matrix Operator Kernel Emissions
SPEED	Average Hourly Speed Inventory Data
SO2	sulfur dioxide

Abbreviation	Expansion
TIR	particulate matter from tire wear
TOG	total organic gases
UAM-AERO	Urban Airshed Model, Aerosols
UAM-IV	Urban Airshed Model, version IV
UAM-V	Urban Airshed Model, Variable grid
UNC	University of North Carolina
UTM	Universal Transverse Mercator
VERDI	Visualization Environment for Rich Data Interpretation
VMT	vehicle miles traveled
VOC	volatile organic compounds
VPOP	Vehicle Population